34 Burnfield Avenue
Toronto, Ontario M6G 1Y5
Canada

Tel: (416) 588-0269 Fax: (416) 588-5641
Web: www.LEAP.com

Leap of Faith Financial Services Inc.

August 14, 2022

ICANN Generic Names Supporting Organization (GNSO)

Subject: Initial Report on the Transfer Policy Review - Phase 1(a) - comments

Submitted by: George Kirikos
Company: Leap of Faith Financial Services Inc.
Website: http://www.leap.com/

Dear ICANN GNSO,

This comment is in response to the call for public comments on "Initial Report on the Transfer Policy Review - Phase 1(a)" as per the notice at:

https://www.icann.org/en/public-comment/proceeding/initial-report-on-the-transfer-policy-review-21-06-2022

Please note that this submission is made **in protest**, given that requests to extend the deadline to mid-September (or even later) were rejected. This is discussed in more detail below.

Sincerely,

George Kirikos

P.S. **Note to ICANN Staff**: Do not butcher this document if/when you paste its sections into your **flawed** public comment review tool. Some sections are interdependent and refer to each other, so might need to be copied/pasted **more than once to be understood in context**!

MEDITATIONS* ON DOMAIN NAME TRANSFERS

by: George Kirikos

* Inspired by Marcus Aurelius.

TABLE OF CONTENTS

"If someone can prove me wrong and show me my mistake in any thought or action, I shall gladly change. I seek the truth, which never harmed anyone: the harm is to persist in one's own self-deception and ignorance." - Marcus Aurelius (from "Meditations")

A. INTRODUCTION

Leap of Faith Financial Services Inc. is a privately held company based in Toronto, Canada. It is the owner of approximately 500 domain names, including school.com, math.com, leap.com, seeds.com, and options.com. This portfolio is worth many millions of dollars. As such, we have a direct interest in any changes to the ICANN transfer policies, to the extent that those changes adversely impact the security of those domain names, the security of domain names that we might acquire in the future from others, and the rights associated with those domain names.

As users of the internet, we are also concerned about the security of **all** domain names, lest the websites and other online services that we rely upon get compromised by attackers. Those compromises can harm us through loss of privacy, loss of service, and other economic (and even non-economic) harms. Often attackers will compromise one system in order to gain access to other systems, so seemingly minor changes in security can have devastatingly large and disproportionate eventual impacts.

We have long been defenders of domain name registrants' fundamental rights in ICANN policymaking, and make our comments in that same spirit in this response to the initial report of the latest working group looking at domain name transfer policies.

This is a **bad report**. Let's not sugarcoat it or pretend otherwise. While there are *some* elements that are positive, they are outweighed by the negative. **The public would be better served if all the output of the working group to date was simply discarded, rather than proceed with their current recommendations.**

Since we doubt the ability of the working group to evaluate their own work in a self-critical manner, or to handle outside criticism very well, we expect

that they will not heed the analysis that comes from the comments submitted by the public, especially if that analysis disagrees with the report's recommendations. We've seen this happen time and time again in ICANN public comment periods. Indeed, that's what **many people** have told us, that they are reluctant to even submit comments, because **they know that they will be ignored**.

Why do we not follow their example, and simply walk away? It would reward ICANN and its insiders if well-informed stakeholders like us said nothing at all. Our 'silence' would be portrayed as 'consent' to ICANN's actions. In our view, ICANN's attempts to shut out (both directly and indirectly) meaningful public input will eventually be its undoing, and comments from critics like our company that have long been on the record will at least be able to demonstrate that ICANN was warned about its bad decisions and policy outcomes. Hopefully some of those who are responsible for ICANN's bad decisions will one day be held accountable in the future by a higher authority.

While we wait for that day to come, we submit these comments in **good faith**, **pretending that they will be fairly evaluated and considered**.

The Roman Emperor Marcus Aurelius made a series of personal writings. They were never intended for publication. Our own "notes" on the domain transfer system and the working group's report are similarly very rough in places, and due to the unreasonable deadline (which was not extended to mid-September as requested) are unpolished. We could have used the extra month to reorganize, restructure and condense the material. We trust the readers to be understanding in that regard, that this is really a "draft" that was forced to go to publication due to time constraints.

B. WORKING GROUP SUFFERED FROM UNBALANCED AND UNREPRESENTATIVE PARTICIPATION

To understand what led to the current report, one need only look at the composition of the working group (see pp. 46-47 of the report). It is **overwhelmingly dominated** by registrar interests, membership and participation. While not directly evident from the report, even the chair is from a registrar (GoDaddy). Domain name registrants are severely underrepresented.

To the extent that Zak Muscovitch is participating, he is constrained by the views of the Business Constituency (really the "IP Constituency" lite, given the huge overlap in their typical policy positions), rather than the true views of domain name registrants themselves or even those of the Internet Commerce Association.

Domain name registrants do not have the identical interests as registrars. When a working group lacks balanced representation, the outcome cannot be trusted to balance the views of affected stakeholder interests. The availability of a public comment period does not cure or ameliorate that condition. Indeed, the imbalance will continue as those comments are reviewed by an unbalanced working group. Only **direct** and balanced representation on the working group itself can even begin to fix this problem.

This is a problem that we've pointed out **repeatedly** (to the public via our blog, and also to ICANN in past comment periods). For example, the RPM PDP was dominated by pro-complainant participation:

https://freespeech.com/2020/04/16/icann-rpm-pdp-phase-1-comment-period-is-another-sham-part-2/

The recent IGO working group showed even more shocking unbalanced participation:

https://freespeech.com/2021/10/20/unbalanced-icann-working-group-participation-harms-domain-name-owners/

https://itp.cdn.icann.org/forms/publiccomment/submission/LEAP-comments-IGO-ePDP-2021-final-20211023.pdf (see pp. 27-30)

While we did not have sufficient opportunity on this occasion to generate the same metrics as in the IGO working group participation analysis, **ICANN should be able to generate those same metrics automatically**, given

that all meetings are recorded (and transcribed). All mailing list messages are also archived.

This is a failure of the bottom-up multistakeholder model. ICANN and the GNSO should:

(i) Do greater outreach even after the comment period has concluded, so that affected stakeholders become aware that proposals exist that will negatively affect their domain names.
(ii) Consider a second comment period that is more widely publicized and longer, to ensure greater opportunity for outreach and study.
(iii) Expand membership of the working group, to ensure that the voice of domain name registrants is heard.
(iv) Rethink the entire restricted membership working group model, which has repeatedly led to these kinds of results.

C. DATA PROBLEMS

Evidence-based policymaking requires good data. Without it, one is acting blindly, without the metrics to properly understand problems, let alone identify solutions whose benefits outweigh costs.

ICANN only pays lip service to data requirements and data collection that are required for evidence-based policymaking. This is self-evident in the report itself, which just ignores the problem for the most part. While section 2.6 mentions data, the report does not show how the recommendations actually flow from that data, and are justified by any data.

Indeed, the truth is evident in the working group discussions themselves, where members openly decry the lack of data. For example, in the June 28, 2022 transcript, page 16, one member says (in the context of domain thefts):

https://gnso.icann.org/sites/default/files/policy/2022/transcript/transcript-gnso-tpr-pdp-28jun22-en.pdf

> **"First of all, we don't have the numbers. And we didn't have the numbers in 2015. And we still don't have them."**

Rather than rely on 'gut', or anecdotes, or simply throwing caution to the wind to 'do something', **there's a path that they ignore, namely going out and actually getting the data!** If you want to take a scientific approach, then you need the numbers, the actual data. Data collection is hard. It can be expensive. It can take time. But, that's how you do things right. There's the "right way" to do things, "the wrong way", and there's "the ICANN way". Unfortunately, the "ICANN way" rarely overlaps or coincides with the "right way."

This is also related to the composition of the working group. If there was a larger and diverse skillset to draw upon (including statistics or quantitative backgrounds), then greater attention would have been paid to pointing out the absence of evidence-based policymaking throughout the report, which is a product of what happened throughout the deliberations of the group.

Without a strong foundation of good data, the work will suffer, and in fact the report **does** suffer.

The solution is not to just 'throw caution to the wind' and proceed regardless. The solution is to **go back and do things properly**, and actually **collect the data**. **Otherwise, the working group itself is all**

"just for show", with a predetermined outcome that cannot be influenced by factual evidence.

D. XPRIZE-STYLE COMPETITION TO IMPROVE DOMAIN NAME TRANSFER SECURITY

Given the scant attention paid to security by the working group, we believe a new approach is needed. Domain name security, including security of the transfer process, is important enough that it calls for **fresh ideas**. We propose that ICANN issue a widely publicized and open "**Call For Papers**" or a competition of some sort, like the "**XPRIZE**" but for domain name transfer and security procedures. This would encourage academics, security researchers, security practitioners, "white hats", penetration testers, and others to take a deeper dive into the domain name transfer system. They would be encouraged and invited to come up with new ideas that would improve security of hundreds of millions of domain names, which are at the foundation of the multi-trillion dollar online economy.

ICANN agreed to receive a controversial **$20 million** from Verisign upon renewal of the dot-com contract. It was intended to improve security.

https://www.theregister.com/2020/01/07/icann_verisign_fees/

https://www.icann.org/en/blogs/details/icann-decides-on-com-amendment-and-proposed-binding-letter-of-intent-between-icann-and-verisign-27-3-2020-en

We suggest that a portion of it, perhaps **$250,000 to $500,000**, be used to fund the total prizes and/or honoraria for an XPRIZE-style competition or call for papers. This is a small fraction of the $20 million.

Such funding would provide an economic incentive to draw new ideas and new eyeballs into the ICANN ecosystem, particularly from academia, rather than from "the usual suspects" who've dominated ICANN for the past two decades. Transfer security, and overall domain name security, is too important an issue to leave to those 'usual suspects'.

[To make it clear that the author of this comment submission would personally not financially benefit from such a competition, folks should be able to have any prizes/honoraria be directed to charities, rather than to themselves, as we would do to eliminate any conflicts of interest that might be seen from making this proposal.]

E. BREAKTHROUGH PROPOSAL: GENERATE DOMAIN NAME TRANSFER TRANSACTION ID AT GAINING REGISTRAR TO INPUT AT LOSING REGISTRAR

Consider this section of our comment submission to arguably be the most important. As we discuss in other sections below, there are considerable weaknesses and disadvantages to the current domain transfer approach, which generates the "AuthInfo Code" (to be renamed "TAC" - transfer authorization code) at the **losing registrar, which is later submitted to the gaining registrar.**

By doing so, there is **too much value** placed on knowledge and/or control of the TAC. If the TAC is compromised/misused by an attacker after it is generated (but before it is used by the rightful registrant at the correct gaining registrar), it's game over. The working group, as discussed later, spent considerable time focusing on the TAC, like its complexity, and when it's generated, and so on. Despite this focus, they never really deeply examined or questioned its place as a "solution", always assuming that it was the right approach.

We believe that a different approach to devalue and deprecate the importance of the TAC, would **improve security while maintaining ease of use**.

We would **entirely eliminate the TAC for the purposes of transfer authentication/authorization purposes** (we understand that the AuthInfo Code is sometimes used for non-transfer reasons; conceivably it can be retained for non-transfer purposes, but no longer be used for transfers).

Instead, we would do transfers in the following manner:

Step 1: Go to **gaining registrar** and initiate a transfer. The gaining registrar provides the intended registrant (this could even be a change of registrant) with a unique transaction ID, which we will call the "**Pending Transfer ID**" or "**PTID**" since the ICANN world loves acronyms (and no one has used that one before!). The PTID should be provided **securely** and **immediately** (e.g. directly on the website when placing an order) to prevent an attacker from injecting a competing (rogue) PTID that would trick a registrant (e.g. if delivery of the PTID was by email, conceivably an attacker could trick a registrant by sending a competing email with the PTID for a competing fraudulent transfer) by spoofing the gaining registrar.

To be precise, the PTID need only be **unique for that domain name**, but

does **not** need to be fancy or encrypted or have special characters, etc., because (as you'll see below), **knowledge of this code is completely worthless to an attacker, by design**!

To ensure uniqueness, **it should probably be generated by the registry**. But, it could just as easily be generated by the registrar (but then the registry could reject it if it's already in use by a potentially competing transfer for the same domain name).

So, to be clear, the PTID could **literally just be a single character**, i.e.:

A

or a number, like:

123456

But, for ease of use and transparency purposes (this is getting into the 'implementation weeds' a bit, but just to demonstrate that there has been considerable thought about this), and to make it more understandable for registrants (**especially if they want to later reject an unauthorized pending transfer**), it should be of the form:

[GAININGREGISTRAR]:[Domain]:[number]

So, something like:

GODADDY:EXAMPLE.COM:8675309

(one could use IANA registrar ID numbers instead of a textual representation of the gaining registrar; if that was done, then the losing registrar should, as a best practice, parse and convert IANA numbers into the corresponding textual representation (and display that textual representation to the registrant) so that the registrant can more easily understand the intended destination of the domain name)

[aside: To score extra technical 'brownie points', there could also be a 2 or 3 digit "checksum" at the end, to detect typos, see:

https://en.wikipedia.org/wiki/Checksum
https://en.wikipedia.org/wiki/Luhn_algorithm

This could be added to the TAC too, by the way, although as we're arguing here, the **TAC should be eliminated**, not kept! (not going to get into which

algorithm to use; the above are just starting points to start learning)

Step 2: The registrant now **goes to the losing registrar**, logs into a control panel (or whatever), and says they want to complete a transfer. **They then can simply input the PTID from Step 1**. The losing registrar would submit it to the registry, and the registry can verify whether or not that PTID exists.

An alternative implementation would be for the losing registrar to simply query the registry for a **list** of all pending transfer requests/PTIDs for the relevant domain name, and allow the registrant to pick the right one to 'approve'. [optionally, the registrant could also directly mark abusive/fraudulent PTIDs, just like they would mark a "spam" email in an inbox, to cancel them (without full refund to the attacker, see below)]

As you can infer, **we've made the knowledge of the actual PTID completely worthless to an attacker**! If the PTID is "compromised", it cannot be taken by an attacker to a different gaining registrar, to "complete" an unauthorized transfer. So, "secrecy" and "control" of the AuthInfo Code / TAC is no longer a concern.

Step 3: (optional, but obviously **we want to retain this**) The ACK/NACK "Losing FOA" step would be retained, for confirmation of the transfer via demonstrable control of the registrant email. [we'd go further, and change the default behaviour so that if there's no response to ACK/NACK at all, the transfer fails, rather than defaulting to success]. This step acts as a last layer of defence if the control panel access is compromised, or other attack scenarios (e.g. rogue employees, 0-day attack on the registrar, etc.)

That's it! Since the knowledge of the PTID is worthless to an attacker, they have to **change the form of their attacks**.

What the bad guys would need to do is:

(a) initiate their own competing transfer at a different registrar,

AND

(b) convince/trick the registrant to use THAT rogue PTID, instead of the correct one.

So, if the "correct" PTID is: GODADDY:EXAMPLE.COM:8675309 but the attacker's PTID is ALIBABA:EXAMPLE.COM:96711111, not too many registrants are going to be fooled.

(of course, this won't stop all security threats, e.g. if the control panel access at the losing registrar is compromised, rogue employees, etc,)

How do we "harden" the system to prevent this new form of attack? As a best practice, we should retain the ability to enable/disable "transfer lock" (i.e. only allow the PTIDs to be generated when the "clientTransferProhibited" is not on, i.e. when the status is "OK"). [**We don't necessarily require the transfer lock**, in theory, because one can no longer "brute force" the AuthInfo Code/TAC, as it doesn't work anymore!] But, **retaining the transfer lock capability would have some value**, to the extent that it **prevents a flood of competing PTIDs** to be generated by attackers. [Indeed, this would be a great opportunity for data collection for research, as registry operators might share data on malicious/bogus PTIDs with ICANN, and conceivably also the current registrar (losing registrar) and even perhaps current registrants.]

Registry operator systems would need to be modified to store **multiple** competing transfer requests (the one true one, and any bogus ones, since the registry would not be able to know which one was the authorized request). One would want to ensure that bogus requests do not act as a denial-of-service attack (blocking valid requests), although the number of pending transfer requests could be used to trigger enhanced security and verification, on a good faith basis. [e.g. if there were 2 pending transfer requests, that might trigger greater scrutiny; if there were 10 pending transfer requests for that one domain name, it should be an obvious sign of an attack]

To prevent abuse, **we would strongly recommend that invalid transfer requests not be refunded, or at best only partially reimbursed**. This would **ensure a direct economic cost is imposed on attackers** who make fraudulent/bad faith transfer attempts. (if you make transfer requests "free" until they succeed, that would simply encourage abusive attacks) [the money from bogus transfer requests could either go to a security fund, or to the registry, or even to the registrant]

One (i.e. the registry on its own, or via ICANN policy) could also optionally add a TTL (time-to-live) to the PTID, so it can only be used for a couple of days, at most, or a couple of weeks, etc. (just to unclog the system; remember, they have no value to an attacker!)

We believe it should be fairly obvious that this new approach **enhances security considerably**, as it **eliminates an entire class of attacks that exist today**, namely attempting to compromise or gain knowledge of the

AuthInfo Code / TAC. Under this approach, the PTID can even be public, can be placed on a billboard, can be put into a written contract (for domain name purchases/sales), can be shared during an escrow process without fear of misuse, and provides an **audit trail for transfers**.

Adoption of this new approach would be a game-changer, to enhance security for domain name transfers.

At worst, it's as good as the current system (since all the emphasis today on security as "proof of control" involves access to a domain's control panel, and optionally the registrant email for the domain via the Losing FOA, which we strongly believe should be retained, as will be discussed later). But, in the typical case, and in the "best case", it's much better than what we have today, since **it's a lot simpler and easier to prevent a "rogue PTID" to be entered into the losing registrar's control panel, than it is to prevent unauthorized use of the EPP AuthInfo Code / TAC** (i.e. that's why all the elaborate security apparatus and focus has been on attempting to enhance the TAC's security, etc.).

Conceivably, this approach could also work **in parallel** with the current system, not having to replace it completely. This would allow for a **transition period**. [although, optionally, registrants should be allowed to opt-in to higher security, so that only the newer approach was permitted for their domain names, once sufficient adoption has taken place by registrars and registries] For those registrars that use the AuthInfo Code / TAC for "Fast Transfer" purposes with various secondary marketplace, a transition period would be helpful to them. It should be clear that "Fast Transfer" would certainly be easy to implement and compatible using this new approach (i.e. the gaining registrar could have side-agreements with registrants/registrars, to allow automated PTIDs as needed, that bypass the 'usual' approach -- i.e. making the losing registrar an **automated agent of the registrant**, who'd submit the PTID to the registry that was provided by the gaining registrar).

Furthermore, it should be **relatively obvious** to keen observers and practitioners that it would be **straightforward to extend the PTID approach to enable bulk transfers of multiple domain names** (i.e. a group of domain names at a single losing registrar, to a new gaining registrar, with the group of domain names sharing a single back-end registry operator). As a registrant of a relatively small portfolio of domain names, we do not require a bulk transfer capability, but it would be easy to extend this approach to that use case, since the PTIDs could refer in theory to a group or list of domain names, instead of a single domain name (although, there'd need to be checks and confirmations by the losing registrar, to ensure that

all the requested domains had the same owner, etc. -- the potential for danger and errors should not be underestimated!). The losing registrar would need to query the registry for the full list of domains requested in the bulk transfer, for example, and present that to the registrant in their user interface. Since our time to produce these comments was very limited, we will not expend the time and energy to elaborate on this use case (if the working group wishes to followup on this with us, they know how to reach us, but as noted above it should be pretty obvious how to do so, to anyone technically skilled).

Another advantage of our proposed approach is that nearly all the "evidence of a crime" will be held by the losing registrar, in the event of a fraudulent/unauthorized transfer. Since the "critical events" take place at the **losing registrar** (i.e. submission of the PTID, and the losing FOA), they will have relevant IP addresses in their logs to perform a forensic analysis, without having to rely on the gaining registrar, who may not be cooperative and who also may be in a different (and potentially unfriendly) legal jurisdiction. Given the original registrant (potential crime victim) knowingly chose the losing registrar (and its jurisdiction), this is a desirable situation. Contrast this with the current approach (and also the proposed model of the working group), where the critical events take place at the **gaining registrar** (i.e. submission of the TAC), and where **loss of control of the TAC** could have taken place anywhere (in time and space) between its generation at the losing registrar and its eventual use (or misuse) at the gaining registrar. Unlike the TAC, which registrars are told not to keep copies of, the PTID can be logged and saved by the losing registrar for potential forensic analysis or for an audit trail, and can keep it in plain text and observable (i.e. it has zero value to an attacker!).

Had we been provided with the reasonable amount of time we'd requested for comments, we could have also produced fancy flowcharts/graphs to compare with the ones in the ICANN report, for a detailed step-by-step workflow or "swim lane diagram". We leave that as an exercise for the working group and/or ICANN's army of paid staff (11 of which appear to have been allocated to this working group, according to page 47 of the report).

A keen observer might be thinking "**What you're proposing sounds a lot like pushing a domain name from the losing registrar to the gaining registrar. Why not just create a public "domain wallet address" to allow for a direct push from the losing registrar to the gaining registrar?**"

Indeed, that's how many registrars handle **internal** transfers within their

registrars. A registrant can simply push a domain name to the account of another registrant.

Similarly, bank wire transfers have the funds pushed from the losing bank to the gaining bank, via SWIFT or Fedwire or with IBAN numbers. In those cases, there's a public account number and destination bank. There aren't any "secret codes" embedded in the wire transfers itself, yet they're secure transfers.

In the crypto space, users also have public wallets, where bitcoin or NFTs or other crypto assets can be pushed to the public wallet address.

So, why not do the same for domain names, and allow that kind of push?

NOT RECOMMENDED PUSH PROPOSAL VIA DOMAIN WALLETS

(you'll note I highlighted this all in RED TEXT, in case it doesn't show up if/when ICANN staff copy it to their Public Comment Review Tool)

I do not recommend this, but let's follow it through:

Step 1: Go to gaining registrar and create an account. Registrar creates a domain wallet address.

Step 2: Go to losing registrar, and "push" the domain to the wallet address in Step 1.

Step 3: (optional, but we would always want it) Retain the Losing FOA step, as a last line of defence against unauthorized transfers, to be able to ACK/NACK transfers.

What's wrong with the above domain wallet? It sounds so great! There are at least 2 big problems:

Problem #1: Folks will send domains that are unwanted! For wire transfers, you don't mind if folks send you money to your public account info! But, as we've seen with crypto, folks will be happy to send pornographic NFTs, altcoins, and other "unwanted" assets to a wallet address. [e.g. folks who wish to hype a certain altcoin or NFT will push it to a celebrity's wallet, to try to make it seem like others are buying that asset] In the context of domain names, bad actors would push TM-infringing domains, pornographic domains, CSAM domains, terrorist-related domains, and other unwanted or abusive domains to a domain wallet. [relying on a wallet being 'secret' won't

work; as an attacker might find out about it]

Problem #2: For crypto and wire transfer recipients, there's no cost to receiving the assets that are transferred (well, technically you might be charged a fee by your bank to receive a transfer, but it would obviously be lower than the amount of the funds that were received, so an attacker can't reduce your balance by sending you a wire transfer). In the ICANN world, when there's a transfer between registrars, the registrant at the gaining registrar has to pay fees for an additional year's renewal.

So, to solve problems #1 and #2, we need a wallet address that shows that the registrant at the gaining registrar (a) wants to **receive only that specific domain**, and (b) **agrees to pay for the fees**!

So, we need a one-time use / ephemeral / temporary domain wallet address for a specific domain transfer, that can't be used by attackers to send us other unwanted stuff, and is generated after we agree to pay for that transfer.

Now go back to the beginning of this section, with our actual proposal and the steps involved -- that's **exactly what we've proposed**!

The PTID is generated after we've paid for the transfer request at the gaining registrar (solving problem #2). And, it's of no use to an attacker who learns of it and who wants to send us other junk -- it's only useful for a particular domain name that we wish to receive at the gaining registrar (solving problem #1).

This is the domain "push" between registrars that folks have wanted for a long time, that actually avoids the 2 problems above. It's secure by design. It matches the payment issues and transactional methods that registrars and ICANN are used to (unlike general wallets in other fields).

In conclusion, we hope to see this game-changing approach seriously considered and adopted by the working group. **It would greatly simplify the group of recommendations, reduce the attack surface for unauthorized transfers, and thereby greatly enhance domain transfer security, while maintaining ease of use for registrants, registrars and registry operators.** Bad actors (i.e. domain name thieves and hijackers) would be **greatly upset** if this new approach was adopted, as they would have a harder time stealing domain names.

F. "THE BEST OF BOTH WORLDS" PROPOSAL TO RETAIN THE LOSING FOA ON AN OPT-IN BASIS BY REGISTRANTS

Consider this section of our comment submission to arguably be the second most important. As will be discussed later on in this comment submission, we are **strongly opposed to the removal of the Losing FOA step**, as it is an **important safeguard**. We vehemently disagree that recommendations #7 through #13 make up for the loss of this important safeguard. The working group, by "holding the pen" on policymaking, presents the community with a false dilemma, namely a choice between today's transfer system or their faster (but less secure) alternative.

Instead, we believe that we can have "the best of both worlds", offering registrants the **choice** of whether they wish to do transfers with or without a Losing FOA. This choice can be enabled for both the transfer system proposed by the working group, but **also for the "Breakthrough Proposal" above** (where the PTID is generated at the gaining registrar to be submitted at the losing registrar, rather than the AuthInfo Code/TAC being generated by the losing registrar for use at the gaining registrar).

Briefly (using the language of the working group, but easily changed for the superior Breakthrough Proposal):

(i) At the time that the TAC is generated (**or even better, set it at an account level at the registrar, which can only be modified by an out-of-band verification, and with a delay if changed to a weaker state, for safety**), give the registrant the **choice**, whether they want the transfer to be "**SuperFast**" (or you can call it "Normal"), or "**SuperSecure**" (Slower).

(ii) If they pick "SuperFast", there'd be no "ACK/NACK" step after the TAC is used at the gaining registrar — transfer would complete immediately (what the new working group recommended).

(iii) If they pick "SuperSecure", there'd be the current "ACK/NACK" step after the TAC is used at the gaining registrar — which registrars already have code for — it's what we have now! This is all automated, too, so it's super-trivial and cheap. [This has greater security in the event the TAC is compromised after it's generated.]

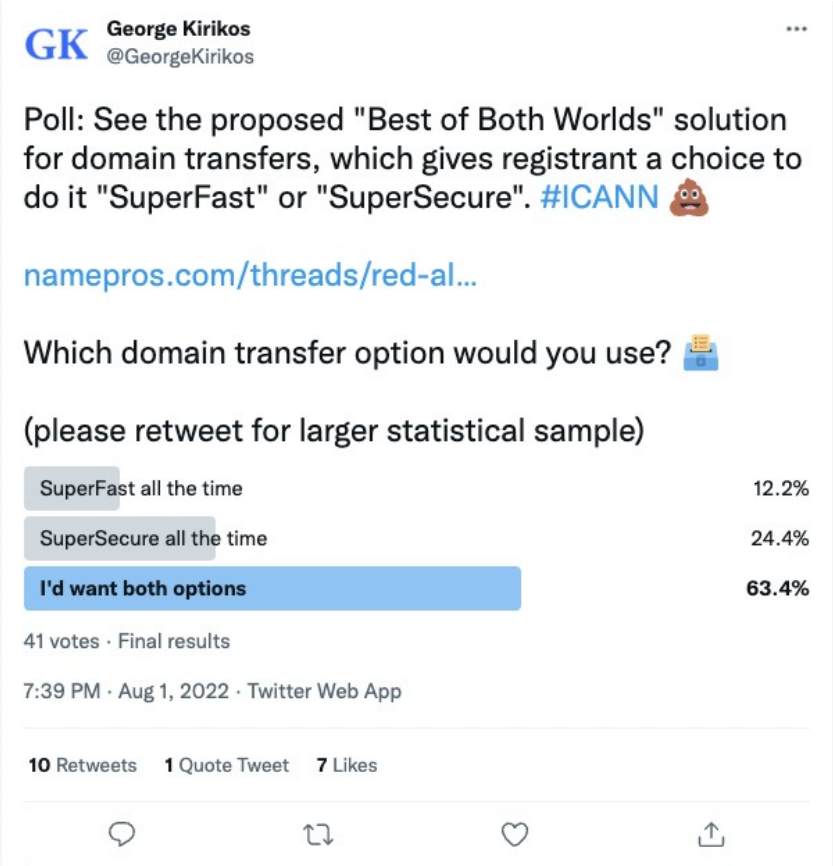
There'd need to be a few pieces of extra code at the registrar/registry to handle the different branches, depending on the path initially set. [but, since the report already proposed a change, they'd be writing new code anyhow!]

So, we can have the "best of both worlds." We think it'd be essential for the security-conscious registrant to be able to make a setting that ALL of their transfers are "SuperSecure" by default, which can't be changed except via out-of-band communications, etc. That preserves our multi-factor security, and also protects against rogue employees somewhat, or if the registrar gets hacked, to some degree. [i.e. if an attacker can override that setting without our consent by breaching the registrar, they can boost their chances of a successful theft]

Thing is, there's always going to be scenarios where a rogue employee at a registrar does something, or there's a 0-day Linux vulnerability that hacks the registrar, or there are some other scenarios where the ACK/NACK saves one's bacon….so we're reluctant to give it up easily. (not all scenarios; i.e. the ACK/NACK step might get hacked at the registrar too — in an ideal security design, strict access control would be taken into account, to minimize the odds of that kind of attack, with lots of logging, too)

We ran a Twitter poll, and you can see the results below.

https://twitter.com/GeorgeKirikos/status/1554250422747074560



(the image above is a static screenshot, instead of an embedded live view, given it's in a PDF submission)

The link to the referenced NamePros thread is:

https://www.namepros.com/threads/red-alert-icann-working-group-wants-to-make-it-easier-to-hijack-domain-names.1279715/page-2#post-8665292

Of course, it's not a statistically significant sample, or a carefully controlled random sample, but it's at least an attempt to get broader input. Clearly, folks like the idea of having the **choice** of a faster or more secure transfer. They don't want to be forced to be "SuperFast" all the time. (which is what the working group would impose, if their recommendation was adopted)

Furthermore (we didn't poll on this), one could enhance choice of security **even further**, by also providing the option of "**UltraSecure**", which would **require an ACK** for the transfer to succeed. [i.e. it would **change the default behaviour** of 'accept the transfer if the registrant doesn't respond' to become '**reject the transfer** if the registrant doesn't respond to the ACK/NACK request']

We believe that default security behaviour should be **conservative**, and thus should not do "dangerous" things if there is no response from a registrant. The **current default behaviour can be attacked**, for example by **flooding the registrant** with hundreds of thousands of emails, to make it more difficult to identify which email contains the "ACK/NACK" Losing FOA request email from the registrar (i.e. essentially a denial of service attack). A more conservative approach would thwart that attack. Furthermore, not all registrants are monitoring their emails carefully or continuously, and so they can miss the opportunity to NACK an unauthorized transfer attempt if they're on holidays, or if the email ends up in a spam folder, or is otherwise not acted upon. The ACK/NACK email of an unauthorized transfer attempt is unexpected by the registrant --- they were not looking for it to come, and so it's a far different scenario than an authorized transfer attempt that generates the ACK/NACK email (where the registrant is specifically told to monitor their email for that communication).

Valuable data can and should be collected as to the proportion of transfers that are "SuperFast" vs. "SuperSecure" vs. "UltraSecure". Those statistics can guide future ICANN policymaking, as they would reveal the actual preferences of registrants in the speed vs. security debate. Those statistics can be analyzed at a global level, or even at a registrar-by-registrar level.

In conclusion, we believe the "Best Of Both Worlds" approach would be an

excellent compromise, recognizing that there is no "one size fits all" for registrants. Registrants who wish to expedite transfers by removing the Losing FOA step would have that choice. Other registrants who wish to retain stronger protections of the Losing FOA step (and even the "UltraSecure" variation of the Losing FOA step, which at one time historically was the default) would also have that choice.

We are encouraged to see that the Internet Commerce Association appears to support this approach of retaining registrant choice of security measures, (they would have read the outline of our proposal via our public blog posts in advance of this formal comment submission).

As an aside, those who would insist on requiring adoption of the current proposed recommendation (i.e. "SuperFast"), because it's "easier" or "eliminates steps" should be aware the logical extension of that argument would be to not even have registrars at all, but to simply go back to the old days of the Network Solutions monopoly, where there was a single monolithic registrar (which was also the registry). That was "easier" too, not having the "complexity" of multiple registrars, transfers between registrars, etc. Arguably, we're better off today than in those ancient times of 1996, because registrants are now able to choose the best registrar for their needs, as the needs of registrants are not uniform. Just as registrants are able to choose between different registrars, they should also be able to choose between different security levels to handle their transfers. This proposal celebrates that desire to have a choice, rather than have someone else take away that choice for all registrants.

G. IMPROVING THE LOSING FOA BY MAKING VISIBLE THE "BEFORE" AND "AFTER" WHOIS INFORMATION

At present, the losing FOA leaves a lot to be desired when it comes to informing the registrant of the precise ramifications of approving a pending transfer request. **The only information they have is (a) the identity of the gaining registrar, and (b) the timing of the request.**

For some people, those two pieces of limited information are enough to convince them that the transfer request is authorized (or to detect that the transfer is unauthorized, e.g. if the gaining registrar doesn't match their **intended** gaining registrar).

However, we **should be able to do a lot better than this**, to not force registrants to make a "leap of faith", **ignoring potential risks** that the actual transfer is not legitimate. **We want certainty**. What we would like to see is the current WHOIS information (i.e. before the pending transfer is accepted), and the proposed WHOIS information of the pending transfer (i.e. what would happen "after" the pending transfer is accepted). Ideally, this would be side-by-side to allow for an easy comparison. By comparing the "before" and "after", we'd be able to have certainty that the transfer is accurate and authorized.

Due to GDPR and related privacy issues, gaining registrars, registry operators and losing registrars might not want to actually send/receive WHOIS data. Instead, the gaining registrar should be mandated **to ask for the consent of the prospective registrant to DISPLAY the "after WHOIS" on the site of the gaining registrar, for the limited purpose of ensuring security of the transfer (a verification mechanism to the registrant at the losing registrar). This consent can be denied by the prospective registrant.** The gaining registrar would then generate a temporary URL with the "after WHOIS". That URL itself would not contain any private information, but would provide a **LINK** to the "after WHOIS". That URL would be served on systems run by the gaining registrar (so it's not passing information to the registry or the losing registrar).

If there were no privacy issues/GDPR, during the Losing FOA process/screen at the losing registrar, we could have embedded the full "before" and "after" WHOIS on a single page, for full visibility to the registrant of the impact of accepting the transfer request. **Instead, we can at least allow the losing registrar to LINK to the "after WHOIS"** (as the LINK can be passed by the registry operator, for a given transfer request). If the prospective registrant at the gaining registrar provided consent, the full "after" WHOIS

would be visible (on the systems of the gaining registrar). If not, the current registrant at the losing registrar **could make up their own mind, depending on their personal risk preferences**, whether or not to ACK or NACK the pending transfer.

Note that this proposal is fully compatible with both the current transfer system **AND** our "Breakthrough Proposal" (where a pending transaction ID is generated at the gaining registrar for submission to the losing registrar).

Under both systems, attack scenarios exist where an attacker could generate an unauthorized transfer at the authorized new gaining registrar, but in a **different registrant account from the intended account**. **This approach would thwart that attack.**

While some might want to defer discussion of this to a later phase of the transfer work (because of the potential for change of registrant issues), we believe that it should be considered in this phase of the work, as it would provide information to a current registrant who **does not want to change the registrant** (who wants to **prevent** a change of registrant during a transfer). Thus, it's a security enhancement. [it can of course also be used to provide greater certainty when there is a desired change of registrant during a transfer]

As an aside, this could also be implemented through incorporation of the "SSAD" (System for Standardized Access/Disclosure), **supplemented to supply pending ("after") WHOIS, not just current ("before") WHOIS**. (although, that's probably overkill, given the complex access control contemplated by SSAD; we'd want something lightweight like the current WHOIS, just for pending/future/"after" WHOIS info)

In conclusion, this better visibility of the "before" and "after" WHOIS would make for a superior audit trail for transfers, and reduce the ability for attackers to succeed with unauthorized transfer attempts.

H. EMBED GAINING REGISTRAR INTO TAC

[The author thanks Zak Muscovitch, Reg Levy, and Jothan Frakes for discussions related to this proposal.]

The AuthInfo Code, to be renamed TAC ("Transfer Authorization Code") is inherently insecure, and should be deprecated, as we argued in our counter-proposal (in section E of our comment submission, namely the breakthrough proposal, which generates a domain name transfer ID at the gaining registrar, to input at the losing registrar). Multiple recommendations appear in the report to try to strengthen it somehow, in advance of it being generated. But, they all fail to recognize that it's "game over" if it is **obtained by an attacker after it is generated but before it is used by the lawful registrant at their intended gaining registrar**. Whoever possesses the TAC essentially possesses the domain name itself. In a blog post:

https://freespeech.com/2022/08/03/die-hard-opposition-to-reduced-security-for-domain-name-transfers/

we used the metaphor of a "bearer bond" (particularly if the Losing FOA safeguard is eliminated).

In other words, the TAC is a high value target for attackers. Our preferred approach would eliminate its value, as discussed in section E of our submission.

But, in the event that the working group doesn't adopt our section E proposal, the TAC itself can be improved upon by embedding the **intended gaining registrar** into the code itself. One can reserve the first few characters of the TAC (perhaps 6 or 8 characters) for the IANA ID of the intended gaining registrar. Thus, instead of generating a TAC that can be used at any gaining registrar (which is a very large attack surface!), we can restrict the TAC (we can call this a "Restricted TAC" or "RTAC") so that it can only be used at a **single gaining registrar**. [We can call the current TAC the "Unrestricted TAC" or "UTAC", given it can be used anywhere.]

By restricting the TAC's usage to a single gaining registrar, its value as a target is greatly diminished, if its compromised. The attack surface is reduced considerably, compared to a TAC that can be taken to a registrar in any jurisdiction (including "unfriendly" jurisdictions from the registrant's perspective, with different legal or political systems, etc.).

If there's no change of registrant during the change of registrar (i.e. the

registrant is simply switching registrars), the registrant knows exactly where they want to end up, and can generate a RTAC that limits potential damage if it's compromised (at worse, an attacker would have to create an account at that particularly gaining registrar, which the registrant presumably is familiar with in terms of legal and political jurisdiction, and could more easily seek recourse if the RTAC is compromised).

If there's an intended change of registrant (e.g. domain purchase/sale, escrow, etc.), the RTAC would also be very useful, to limit opportunities for fraudulent behaviour from the escrow firm, buyer, and any other attacker. With an unrestricted TAC, the buyer or escrow would have **plausible deniability** if the domain name ended up at a completely different gaining registrar than that specified in a legal agreement/contract. They could argue that they never even received the TAC (even if they did!), that someone else misused it, etc. A restricted TAC would assist considerably (that's another reason the Losing FOA is so important, to know that a domain ended up at the correct gaining registrar, and one could NACK it if it did not).

This RTAC can be created now that the working group has recommended that TAC only be generated only on request (Recommendation #9), whereas previously an AuthInfo Code always existed for a given domain name.

In some sense, this proposal can be considered somewhat related to (but not motivated by) Recommendation #13 (TTL). That recommendation seeks to reduce the "attack surface" of the TAC in the time dimension. In contrast, our proposal seeks to reduce the "attack surface" of the TAC in the dimension of the gaining registrar, by limiting its usefulness to only a single gaining registrar.

In terms of implementation, the IANA code should act as a plaintext prefix to the rest of the TAC. So, for example if the TAC would have been:

TAC: Before (unrestricted): h7-m191MBCH1RCXN

and the IANA code of the gaining registrar is 4321 (currently not used by anyone at the time of this submission!), the Restricted TAC would be: [assuming first 8 characters reserved for the IANA ID]

RTAC: After (restricted): 00004321h7-m191MBCH1RCXN

If one wanted to make it more visible, one could add square brackets, e.g.

RTAC: After (restricted): [00004321]h7-m191MBCH1RCXN

at the cost of adding 2 additional reserved characters at the beginning.

The losing registrar would need to present the registrant with a dropdown listbox (or other user interface element) to select a gaining registrar, if this approach is adopted, as part of the user interface that generates a TAC. The gaining registrar would also likely need to document in their online help their IANA ID, as they instruct customers on how to complete a transfer of domains to them.

Conceivably, one could extend this further, to specify a target reseller within a registrar (although that's probably overkill, in our opinion).

A further enhancement might be to embed a token representing the intended registrant at that gaining registrar, i.e. of the form:

[AAAAAAAA][BBBBBBBB][regular TAC]

where "AAAAAAAA" is the gaining registrar, and "BBBBBBBB" is a token representing the new registrant at that gaining registrar. One could use a similar technology as proposed in section G above to provide a mapping of the proposed new WHOIS of the registrant at the gaining registrar, so that it could be input at the losing registrar when the TAC is generated. An alternative would be to perhaps just embed a (hash) of the email address, or another unique identifier (beware GDPR!), preventing an attacker from using a compromised TAC unless they used that same unique identifier in their WHOIS when creating an account at that intended gaining registrar. Given the largest registry (dot-com) is not thick (registrant info not stored at registry), embedding the intended registrant into the TAC would require enforcement by the gaining registrars, some of whom are perhaps not as trustworthy as others to expect actual enforcement to occur.

For security conscious registrars, they might even create advanced interfaces which permit their clients to designate in advance a "whitelist" of gaining registrars, thereby placing restrictions on the TAC generation tool, reducing the attack surface. This would be a protection in the event that their control panel account was compromised (prevents an attacker from generating a TAC to a registrar that's not on the whitelist), **particularly if changes to the whitelist can only take place on an out-of-band basis**.

Our company, as an example, might specify that the "whitelist" be the "null set" (not allowing the TAC to be generated to **any gaining registrar**) [although, if the "null set" is tough to program for the registrar/registry, we could make the "whitelist" be simply the current registrar, so it could only be transferred to where it already is!] A Fortune 500 company with domain

names at MarkMonitor might whitelist only other "brand-oriented" registrars such as CSC and/or Safenames and/or Com Laude, to limit their exposure to unauthorized transfers. This would enhance security considerably.

I. TIMELOCK ACCESS TO TAC GENERATOR, AKA "VACATION MODE" or "LOCKDOWN MODE"

As noted throughout this report, the TAC is bad. Recopying our text (knowing that our submission will likely be chopped up in the Public Comment Review Tool!)....

The AuthInfo Code, to be renamed TAC ("Transfer Authorization Code") is inherently insecure, and should be deprecated, as we argued in our counter-proposal (in section E of our comment submission, namely the breakthrough proposal, which generates a domain name transfer ID at the gaining registrar, to input at the losing registrar). Multiple recommendations appear in the report to try to strengthen it somehow, in advance of it being generated. But, they all fail to recognize that it's "game over" if it is **obtained by an attacker after it is generated but before it is used by the lawful registrant at their intended gaining registrar**. Whoever possesses the TAC essentially possesses the domain name itself. In a blog post:

https://freespeech.com/2022/08/03/die-hard-opposition-to-reduced-security-for-domain-name-transfers/

we used the metaphor of a "bearer bond" (particularly if the Losing FOA safeguard is eliminated).

In other words, the TAC is a high value target for attackers. Our preferred approach would eliminate its value, as discussed in section E of our submission.

But, in the event that the working group doesn't adopt our section E proposal, we propose that access to the TAC generator itself be restricted through a "Time Lock". We call this "Vacation Mode" or "Lockdown Mode".

If a registrant's account is compromised (or if there is a rogue registrar employee, or a zero-day attack), and a TAC is generated (simply by removing any domain lock, i.e. turning of "clientTransferProhibited"), there is potential for great harm. So, can we restrict access to the TAC generator itself, to reduce the attack surface considerably?

We propose that the registrant be able to restrict access to the TAC generator for a specified amount of time (say up to 30 days maximum), and that **this be enforced by the registry**. Think of it as a timed "serverTransferProhibited", which eventually expires (although a registrant can keep extending it, even before it expires). Because this can be fully

automated, it can be done for **free**. [i.e. the reason why registry lock is expensive is that **out-of-band verification** to **remove a lock** is costly, in terms of requiring human salaries, etc.; you never need out-of-band verification to reapply a lock, which increases security; you only need "human" intervention when you're **decreasing** security, for verification]

So, if we know that we'll be on vacation for 2 weeks, we can put in a 14 day timelock knowing that there's no good reason for us to need to be able to transfer the domain name in that period. We won't have to worry about missing an ACK/NACK email, either. And because this is enforced by the registry, any potential compromise of the registrar or the registrant account would prove **ineffective** in terms of stealing a domain name.

Another way to look at this is that it's a GAPPED TTL (contrast with the TTL proposal in Recommendation #13). i.e. under a TTL, the TAC can be used from the present until some date in the future. Under our proposal, the TAC can only be generated from some **starting date** in the **future**! (i.e. 14+ days away, in the example in the prior paragraph).

Why is this of value? Registrars and registrants can be compromised, so enforcing this at the registry level would make large classes of attacks impossible! For security conscious registrants like our own company, we could literally go on vacation and not have to check our domain names to see whether they were stolen while we were away. Furthermore, we could keep extending things regularly, knowing that we rarely do transfers to other registrars (on the rare occasion we ever did a transfer, it likely would involve a legal contract that would take weeks to negotiate, so a lockdown mode of 10 or 20 days doesn't really affect us, but would improve security immensely.

While this might be seen as a "poor man's registry lock", it's not identical to registry lock. It's only about preventing access to the TAC generator, to block unauthorized transfers more effectively and without requiring human intervention (as out-of-band verification requires for most registry locks). Nameserver changes could still take place, for example.

Recommendation #9.1 (TAC created on request) **enables** our proposal, as previously/currently the AuthInfo Code **always exists** for a given domain. Our proposal **hardens the security of Recommendation #9.1**, and is consistent with, but stronger than, **Recommendation #20** ("deny a particular transfer request, or a general objection to all transfer requests received by the Registrar, either temporarily") as it's **registry-enforced**.

J. AUTHINFO CODE / TAC IS RADIOACTIVE, TOXIC, DANGEROUS LIKE A KEY TO THE KINGDOM

The AuthInfo Code, which the working group wants to rename as the Transfer Authorization Code (TAC) was a very poor design choice, ignoring sound security principles. It's radioactive, toxic and **dangerous**. It's too valuable, and so attackers want to steal it because it is the "key to the kingdom" to determine the fate of a domain name under the current system (and that proposed by the working group).

Let us step back a little and discuss how we've approached our own domain security. We have security of up to **4 factor authentication** protecting our domain names, that an attacker would need to breach in order to succeed with an unauthorized transfer. To access the master control panel, there's 2 factors of security (a password and a second factor). We disable password resets by email for the control panel user. Furthermore, the admin email for domains (where a Losing FOA would be sent) has 2 factors of security also (Google Advanced Protection, password plus hardware security keys, which shouldn't be a surprise, as one can check the MX records of that domain used for admin email and assume we maximize security). A compromise of that admin email for a domain would not give one access to the control panel! To the extent possible, we try to make things as independent as possible (that's why we use 'up to 4' above).

With the Losing FOA kept, we'd still have a layer of defence against an unauthorized transfer (since that admin email protection could still save a domain). Without the Losing FOA, there'd be **no security remaining after a TAC is generated** (security in depth principles are destroyed).

So, that's why we see the TAC as being too valuable for an attacker, and dangerous. We think the working group sees that it is too, because they've tried (but fail) to augment the security of the TAC. It makes recommendations about when it's generated (instead of having it always exist), it makes recommendations about its lifetime (TTL). It makes recommendations about its complexity and composition, and how it should be stored.

But, once it's actually generated, there's a race to be the first to use it! In other words, there's literally no security left once it's provided to (hopefully) the registrant. If it's compromised, all the risk is borne by the registrant. There's no audit trail of what happens after the TAC is generated, i.e. how many people had access to it, especially if it's shared with others in escrow or with a buyer who wants to transfer to another registrar (where internal transfers might not be desirable for tax or legal reasons).

The TAC is like a password, and we know all the problems with password management. We've seen the same for AWS access keys that are stolen. They're all supposed to be kept secret.

It's like a private key in crypto (although there's no actual encryption or private/public key pairs), where you must keep that key a secret.

So, the attack scenarios that the working group seem to focus on are only up to the time that the TAC is generated. The working group ignores the time between the generation of the key and its actual use. That's the window of opportunity where an attacker can succeed, by compromising the TAC.

Just to give a few examples, the registrant's computer might have already been infected by a virus/RAT trojan at the time they received the TAC. Similarly, corporate networks are breached and attackers patiently wait for opportunities to "do something" after they've infiltrated. There can be 0-day attacks against the registrar, or even rogue employees. We don't want to provide a 'how-to' guide here, but just want to stress that there are many ways to steal a high value target, which the TAC represents.

As we discussed in our blog post:

https://freespeech.com/2022/08/03/die-hard-opposition-to-reduced-security-for-domain-name-transfers/

you can have a registrar with 7 layers of security, like the Nakatomi vault in the movie "Die Hard". But, once you've generated the TAC, the domain is essentially transformed into the equivalent of a "bearer bond" with no remaining security.

The working group has focused on security before the TAC is generated, but once it has been generated, the registrant is expected to walk that "bearer bond" to another registrar (another financial institution) at their own risk.

Now, perhaps some folks are willing to do that, as they think "what could go wrong?" They think it's just easy to copy/paste a TAC code between two browsers, and that's the end of the story. But, that's just one use case (and that machine that did the copy/paste might have been infected, and that was the opportunity that an attacker was waiting for, when the defences were down and dropped to zero). The higher the value, the greater the risk.

Shrinking the window for an attack doesn't matter very much these days,

when attacks can be fully automated and instantaneous.

Furthermore, the TAC might be **intentionally shared** with 3rd parties, to complete transactions (escrow, buyers, lawyers, etc.). There, the recipient could fraudulently use the TAC at a different registrar, but then have plausible deniability (since more than one person obviously would have had access to the TAC, i.e. buyer and seller, or there could have been an unknown breach, 0-day, etc.) [this is why the Losing FOA is still important, to have some protection!] [and an internal transfer might have been undesirable for tax/legal reasons]

So, the "solution" isn't to keep doubling down on the TAC approach, as the working group has done. Instead, adopting the approach that we proposed in Section E, where there is no 'valuable secret' like the TAC to protect, by design, would make an enormous improvement. There, the security is maintained at all stages of the transfer, rather than requiring a "leap of faith" at the critical moment.

It's best to eliminate that entire class of attacks in one fell swoop, with a **superior process**, rather than try to fix the TAC which is inherently insecure.

K. DOWN THE RABBIT HOLE, A DEEP DIVE INTO THE LIST OF RECOMMENDATIONS

Although we've indirectly referenced the recommendations in the above sections, we'll also mention each recommendation below.

Note For ICANN Staff: you might need to copy/paste some of the sections above into MULTIPLE recommendations, or at least cite them, as they don't simply refer to one at a time. Had we been given more time, we could have organized this much better!

Rec #1: Support Recommendation intent with wording change

While we agree that the Gaining FOA can be eliminated, we disagree with some of the analysis on page 13 of the report. We disagree with the part that says:

"The provision of the TAC is sufficient confirmation that the RNH intends to transfer the domain, and therefore the Gaining Registrar does not need to request this confirmation via another means."

In our view, the provision of the TAC simply means that *someone* has requested it, but it might not have been the RNH (e.g. social engineering, hacking, etc. We don't agree that Recommendations 3-4 (mentioned earlier on that page) are sufficient, as they are mere notification attempts, which might not have been received or acted upon, and thus are not confirmation of anything.

As our own counterproposal (in Section E, BREAKTHROUGH PROPOSAL: GENERATE DOMAIN NAME TRANSFER TRANSACTION ID AT GAINING REGISTRAR TO INPUT AT LOSING REGISTRAR) shows, we don't need the gaining FOA, as long as we keep the Losing FOA.

So, to summarize, we can support the recommendation, **but none of its analysis**. (this includes disagreeing with answers to all related charter questions)

Rec #2: Significant change required: changing intent and wording

We disagree with **all of the analysis**. The Losing FOA should be retained, as discussed in the above sections of this document.

In particular, we draw your attention to:

F. "THE BEST OF BOTH WORLDS" PROPOSAL TO RETAIN THE LOSING FOA ON AN OPT-IN BASIS BY REGISTRANTS

G. IMPROVING THE LOSING FOA BY MAKING VISIBLE THE "BEFORE" AND "AFTER" WHOIS INFORMATION

although we talk about the Losing FOA's importance in other sections too (including Section E, which would retain it for our own alternate transfer proposal where the transaction ID is generated at the gaining registrar).

Furthermore, the suggestion re: "inconvenience" (on page 17) of a delay is overblown. For our own very rare outgoing transfers, the Losing FOA is typically within 20 minutes.

Delays for critical changes are often **desirable** in a well-designed security architecture. For example, if you enable Google Advanced Protection, and then try to do a Google Takeout (account backup/download), that backup of your account will be delayed by several days (whereas it happens typically within an hour or two on a regular account). Similarly, Authy intentionally adds time delays to account recovery requests, see:

https://authy.com/phones/reset/?proceed=true

So, speed is not the only metric that matters, at the expense of all other considerations.

It's important to consider the data. According to the Transfer Policy Status Report from 2019:

https://www.icann.org/uploads/ckeditor/IRTPPSRRevised_GNSO_Final.pdf

(linked to from page 11 of the initial report), there were approximately **4,968,000 domain transfer per year**, roughly 0.3% of total domain name registrations. Chart 4 on page 23, and the table on page 24, show that "NACKs" are done an average of 12,348 times, vs. an average of 414,000 transfers per month. That means that approximately 1 in every 34 domain name transfers is NACKed!

That means, in the course of a year (12 months), approximately up to** 150,000 domains are saved from unauthorized transfer attempts by the "NACK" system. [** we use "up to", as the data didn't have the granularity to specify the reason for each NACK, since conceivably a NACK might take place for another reason.] Thus, on that basis alone, it's worth retaining as a

process.

If we contrast the "NACK" usage with other ICANN processes and procedures (e.g. the URS, the TDRP, the UDRP), it is a more frequently used protection than those other procedures. It shows the disregard for registrants that ICANN will potentially take away an important safeguard for security-conscious registrants, but keep less frequently used procedures (such as the URS and UDRP) that benefit stakeholders like the trademark holders. That's a double standard. The damage to a domain name owner, and its users, from a stolen domain name can far exceed the damage to a trademark holder from infringing behaviour on a 1 cent .xyz domain.

It's also worth considering how security is handled for similar transfers in other industries. Mobile phone number transfers/thefts are a great comparison. We found a newspaper article in the Globe & Mail titled "Canada saw surge in phone-number fraud in 2019, 2020, figures show" (POSADZKI, ALEXANDRA; page B2, Sept. 29, 2021):

https://www.theglobeandmail.com/business/article-figures-show-frequent-phone-number-fraud-in-canada-in-2019-2020/

The article is paywalled, but we found the print article using a different database. That article notes that there were 21,589 fraudulent customer ports in the Canadian phone number system over a 10 month period from mid-2019 to 2020. These are similar to domain name thefts, but for phone numbers. Roughly 1 percent of transfers were unauthorized (with a peak in a single month of 2.5%). Importantly, according to the article:

> The CRTC recently disclosed the total number of unauthorized ports and SIM swaps **declined by 95 per cent** from October, 2020, to May, 2021, **owing to new security measures undertaken by the carriers.**

95 per cent is an enormous reduction in unauthorized transfers. How'd they do it?

There was an article about how Canadian mobile phone porting requests were made more secure (**presumably leading to the 95 percent reduction in unauthorzed ports and SIM swaps mentioned above**), by adding verification:

https://mobilesyrup.com/2020/11/05/canadian-carriers-implement-new-number-porting-verification-process-to-prevent-fraud/

> The carriers have launched a new mobile number porting system that **requires**

> **customers to respond to an SMS confirmation before porting occurs**.
>
> It essentially offers **additional verification to ensure a request from another provider to transfer a customer's service**. It **also confirms that the telephone number is generated by the customer and not a fraudster. If the customer doesn't confirm the request then the transfer will not take place**. [emphasis added]

This is quite comparable to the "Losing FOA" process for domains! Unlike domains, where a transfer goes through if there's no response to the ACK/NACK email, the cell phone porting out verification is better. By default the phone number won't port out if there's no response. (comparable to the "UltraSecure" approach we discussed in Section F ("THE BEST OF BOTH WORLDS" PROPOSAL TO RETAIN THE LOSING FOA ON AN OPT-IN BASIS BY REGISTRANTS) above.

Here's an example of what that verification looks like, in the Canadian mobile phone system (see the "Transferring Your Public Mobile Number To Another Service Provider" section at the very bottom):

https://www.publicmobile.ca/en/bc/get-help/articles/port-fraud-protection

> Transferring Your Public Mobile Number To Another Service Provider
>
> To help protect our customers from fraud, Public Mobile will send you an SMS text message should we receive a request to transfer your mobile phone number to another carrier. This step is designed to protect your account by confirming if the request is genuine or fraudulent.
>
> The SMS text will read as follows: Public Mobile message: We've received a request to transfer this phone number to another service provider. To approve this request, please reply "Yes". If you did not request this transfer, please reply "No". Please note that you must respond within 90 minutes. If we don't receive a response within this time, the request will be automatically cancelled. For any issues with this number transfer, contact our Porting Team. Thank you.

By reducing security and verification of domain name transfers, ICANN would be making similar attacks on domain names easier, doing the **OPPOSITE** of what Canadian telecoms did to reduce unauthorized phone number transfers.

ARIN's procedures for the transfer of IP addresses can guide us:

https://www.arin.net/resources/registry/transfers/quickguide_transfers.pdf

You'll note that **confirmation** is an essential part of the process. From page 2, "The **source RIR confirms** the authorized resource holder wishes to release the resources **to the specified recipient** within the ARIN region." (emphasis added) So, not only is there confirmation, but they also confirm that the transfer is going to the recipient desired by the current holder. (consistent with our own proposals)

Many legal contracts specify notification by courier and FAX, to improve the odds of actual notice, as opposed to the attempt at notice. We understand some folks mock the FAX machine these days, considering it a relic of the past. We respectfully disagree. It has its place to ensure redundancy in communication methods, in contrast with the working group's proposals, which emphasize speed of completion of transfers above all other considerations (like security and verification).

The working group also seemed to ignore past ICANN Security and Stability Advisory Committte (SSAC) security advisories, which have touched upon various important considerations. Working group participants should refresh their knowledge by re-reading:

SAC040: https://www.icann.org/en/system/files/files/sac-040-en.pdf

SAC044: https://www.icann.org/en/system/files/files/sac-044-en.pdf

SAC074: https://www.icann.org/en/system/files/files/sac-074-en.pdf

for starters. [We understand SSAC isn't participating in the working group, but they should be! Or they should at least monitor it for security issues, instead of leaving it to us to spot the issues for them!]

Without pointing out every concern, consider page 17 of SAC040:

> Change notifications or confirmations. Some organizations protect against unauthorized or erroneous changes by creating a workflow whereby certain actions require confirmations from multiple parties. Multiple confirmations improve an organization's defences against impersonation: an attacker must socially engineer or impersonate not just one party, but two. Certain organizations may be interested in opting into a service where registrars check for and require multiple, unique points of contact. By doing so, such organizations can extend the same kinds of workflows they use internally to encompass changes to points of contact, domain transfers, or DNS configuration. For organizations that do not have such workflows, registrars could offer an optional service to enable such workflows on behalf of the customer. For example, at initial registration a registrar's change confirmation service could check that the customer has submitted a unique point of contact for each required contact associated with the domain. It also could allow the customer to select which

> points of contact must be notified upon a request to change DNS configuration, or require that both the technical and administrative contact respond by phone or email before making a change requested by one party. In addition, change confirmation can help avoid a vindictive or opportunistic domain transfer. Consider, for example, a situation where an employee designated as a point of contact has left the organization and the organization failed to change the contact information from this employee to his replacement. If the employee left disgruntled, he might attempt to claim the domain through a domain transfer. In the change confirmation scenario, other contacts are required to confirm the transfer and the transfer attempt could be blocked.

This emphasizes the need for ***actual confirmation*** (i.e. like the Losing FOA, but stronger, to require an ACK), and it goes even beyond with MULTIPLE confirmations! SSAC should be aghast to see what this working group came up with, compared to SSAC's past advisories!

Page 17 mentions multi-recipient notifications (whereas this working group only wants notifications to a single contact, the RNH).

Indeed, look at page 18!

"Treat transfer attempts as a security event (check and re-check)."

We put the above in an 18 point font, with YELLOW highlighting, for a reason (in case it doesn't copy over in the Public Comment Review Tool, when ICANN staff copy/paste). It really speaks for itself. "Check and re-check" means confirmation, "red alert", not just "notices" after a major change has already been completed. This highlights the need for the Losing FOA to be retained.

We could spend another 10 or 20 pages simply going through all the past SSAC advice, and showing how it was ignored by the working group. But, this document is already longer than the working group's report, and we weren't granted the mid-September deadline that we asked for, either. We will save everyone some time, and just point out how poorly the working group dealt with security, and if you want further details, start re-reading the SSAC advisories above for yourselves in detail.

Lastly, we draw the working group's attention to the bottom of page 53 of the Final Issues Report:

https://gnso.icann.org/sites/default/files/file/field-file-attach/final-issue-

report-pdp-transfer-policy-review-12jan21-en.pdf

which stated:

"This is carried over in the EPDP Phase 1 recommendation 24) Transfer Policy section II.C.1.4 provides that a **registrar must obtain confirmation of a Change of Registrant request from the Prior Registrant, or the Designated Agent of such, using a secure mechanism to confirm that the Prior Registrant and/or their respective Designated Agents have explicitly consented to the Change of Registrant**."

So, eliminating the Losing FOA would actually create an absurdity, that a change of registrant within a registrar is actually **more secure** (i.e. explicit consent to the change was made), compared to the critical change of changing registrars! (even the current Losing FOA doesn't go far enough, if you look at that language for a change of registrar; one would really require the "ACK" as per our "UltraSecure" option in section F ("THE BEST OF BOTH WORLDS" PROPOSAL TO RETAIN THE LOSING FOA ON AN OPT-IN BASIS BY REGISTRANTS).

Rec #3: Significant change required: changing intent and wording

This is completely insufficient. The TAC, as we've argued above (section J. AUTHINFO CODE / TAC IS RADIOACTIVE, TOXIC, DANGEROUS LIKE A KEY TO THE KINGDOM) is a high value target. By design, it should be eliminated (as our counterproposal in section E makes clear). Simply providing notice that it's been given to someone **after it's already been given** is like providing a notice to someone that $5,000 has been taken out of your bank. It's too late, as the money's already gone!

Rec #3.2 literally recognizes that the request might be unauthorized and invalid. Does the working group believe that registrants are glued to their computer 24 hours per day, 7 days per week, waiting to react instantly to notifications?

There is no requirement in this recommendation for **actual notice** to have been obtained **before the TAC is provisioned**. This is an incredibly poor design. One needs to have **affirmative consent** from the registrant **before** the TAC is provisioned. This is the conservative approach.

Attacks can happen on an automated basis. If anything, the attacker already has what they need (the TAC), and so it will be game over if this

recommendation was to be adopted unchanged.

Perhaps the working group can modify the language and intent, to say that there's a notification that the TAC has been **requested** (as opposed to have been provisioned). In which case, the TAC has not yet been supplied to an attacker, and then the RNH has the opportunity to cancel the request (or to go through with it, by logging in using a token provided by that request, and using the token to complete the request -- this honours all multi-factor authentication that a registrant might have put into place, especially if they separated the user control panel access from their domain name email access (which is advisable, if a registrar has designed things well!).

But, from the language of 3.2, it seems that it's **already** been **provided** to the attacker, not just requested.

This is a very dangerously designed approach, as currently worded, and it really highlights how dangerous the TAC itself is....we can see the working group is **attempting** (but failing) to reduce the danger, but this isn't even close to enough. People miss emails, people aren't monitoring things closely, especially when they're not expecting a notification to begin with.

On the scale of "criticality", provisioning of the TAC should be a "red alert", "danger", etc. It's an unusual event that demands more than just notice that you've given up "the key to the kingdom." It's not like a notice that a renewal has been made, or something similarly innocuous.

Also, **notices** should be sent to all contacts (including the tech contact) not just the RNH. [actual TAC should be given only to RNH, but maximizing actual notice opportunity can occur when multiple contact points are touched]

The better way, as we strongly argue in Section E, is to make slight modifications so that you can **completely eliminate the TAC itself** (the existence of the TAC is the root cause of so many problems/concerns, as can be seen by the various recommendations!).

Rec #4: Significant change required: changing intent and wording

While we support the Losing Registrar sending a notice that the transfer has completed, it should be without delay (rec #3 already mentioned a 10 minute standard).

Also, **notices** should be sent to all contacts (including the tech contact) not

just the RNH. A tech contact might be in a superior position to detect loss of services impacted by an unauthorized change of registrar (e.g. nameservers that used to be the losing registrar might stop serving results once the transfer is complete), and link them to the change of registrar.

However, this notice is not of any "value" whatsoever, in terms of justification for eliminating the Losing FOA, for example. Since by this point the "attacker" (for unauthorized transfer) has won!

By the way, the ID of the gaining registrar should always be in the Losing FOA (as it currently is).

Furthermore, we are strongly opposed to the implication that the working group is trying to revive the previously rejected ETRP (Expedited Transfer Reversal Procedure), from IRTP-B. That was a very controversial proposal that was soundly rejected after considerable debate. See the public comments at:

https://forum.icann.org/lists/irtp-b-initial-report/index.html

and we led the opposition against that. It would have a **profound negative impact** on the secondary market for domain names if there was uncertain title over domains, with the ability of those with seller's remorse or even fraudsters to simply undo a **legitimate** transfer. It would degrade the entire asset class (because one would have to factor in all the potential friction/legal costs to challenge the inappropriate use of the "undo", and so the mere existence of that procedure would cause domain asset prices to decline, even if folks committed to never wanting to use it for themselves! The fact that others could use it means it would hurt everyone.

It appears to us that the working group is **intentionally lowering security standards** in order to **require a way to undo unauthorized transfers**.

Furthermore, an "undo" of a transfer, while it might transfer back control of a domain name, completely ignores the **immense damage** that can take place in a short time when a domain name is hijacked (e.g. resetting passwords of linked 3rd party accounts by hijacking emails, e.g. bank accounts, crypto accounts, social media, etc.; network intrusions, installation of ransomware, stealing data of millions of people, blackmailing people if personal data is stolen, etc.). [We've seen that all the time when hijacked cell phone numbers are used to liquidate bank and crypto holdings, or used to infiltrate corporate networks, etc.]

For charter question a9 "For example, should affirmative consent to the

Losing FOA be considered as a measure of additional protection?"

please note that our own counterproposal in section F. "THE BEST OF BOTH WORLDS" PROPOSAL TO RETAIN THE LOSING FOA ON AN OPT-IN BASIS BY REGISTRANTS had the "UltraSecure" option, which would allow for affirmative consent to the Losing FOA on an **opt-in basis** for registrants.

Question to the community: Should the Gaining Registrar's IANA ID be provided by the Registry Operator to the Losing Registrar so that it may be included in the Notification of Transfer Completion sent by the Losing Registrar to the Registered Name Holder? Why or why not? Please explain.

Yes, it should be provided, along with the long form of the gaining registrar's name, with both also in the Losing FOA (which should be retained, as we've noted in a prior question/section). The gaining registrar is public in the WHOIS, and has no right to privacy or anything. If some registry operators are "broken" and are using an internal "client ID", they should fix their systems.

Rec #5: Significant change required: changing intent and wording

The TAC, as we've argued above (section J. AUTHINFO CODE / TAC IS RADIOACTIVE, TOXIC, DANGEROUS LIKE A KEY TO THE KINGDOM) is a high value target. By design, it should be eliminated (as our counterproposal in section E makes clear).

Calling it the "TAC" is like calling it the "Fluffy Bunny" -- it seems safe, innocuous, with nothing to be concerned about.

If ICANN actually called it "Plutonium" or "Private Key" everywhere in the document, instead of "TAC", perhaps the public would more easily realize that it needs to be handled with extreme care, and indeed should not even exist (as our counterproposal makes clear can be accomplished).

Rec #6: Significant change required: changing intent and wording

The TAC, as we've argued above (section J. AUTHINFO CODE / TAC IS RADIOACTIVE, TOXIC, DANGEROUS LIKE A KEY TO THE KINGDOM) is a high value target. By design, it should be eliminated (as our counterproposal in section E makes clear).

Calling it the "TAC" is like calling it the "Fluffy Bunny" -- it seems safe, innocuous, with nothing to be concerned about.

If ICANN actually called it "Plutonium" or "Private Key" everywhere in the document, instead of "TAC", perhaps the public would more easily realize that it needs to be handled with extreme care, and indeed should not even exist (as our counterproposal makes clear can be accomplished).

Furthermore, conceivably the TAC can be **generated by the registry**, not just the registrar.

Lastly, while the generation of the TAC **on request** (as opposed to always existing, like the current AuthInfo code) is a slight improvement, it is overstated as a huge improvement, because the existence of a domain name transfer lock blocks the persistent (always existing) AuthInfo code.

Rec #7: Significant change required: changing intent and wording

See our prior discussion in our answers to Rec #5 and #6, especially: section J. AUTHINFO CODE / TAC IS RADIOACTIVE, TOXIC, DANGEROUS LIKE A KEY TO THE KINGDOM) is a high value target. By design, it should be eliminated (as our counterproposal in section E makes clear).

But, let's take this further. Rec #7 focuses on the complexity of the TAC, its length, etc. As we strongly argue in the other sections noted above, this complexity prevents a small class of attacks (i.e. brute force guessing of the TAC). It does nothing if the TAC itself is compromised between the time it's generated and before it is used by the rightful registrant at the intended gaining registrar. That complexity does nothing to stop a whole slew of attack scenarios, as discussed above.

The problem is in the TAC itself being such a high value target that needs to be kept secret, not the length of the secret, etc. In contrast, our counterproposal in section E doesn't rely on the PTID (pending transfer ID) being complex -- it need only be unique for that domain. It can even be public, and a single character, and yet be **stronger** (due to the different process design) in terms of overall security.

Furthermore, we take issue with some of the guidance in Section 4.1 of the RFC 9154, which mentions printable ASCII character and also case insensitive characters. One might find it advisable to reduce the group of permitted characters even further, to eliminate those that look alike (e.g. zero (0) and capital "o" (O), and lowercase L (l) and capital "i" (I) and the digit "one" (1), depending on fonts). One might want to also pay attention to how those letters/numbers/printable characters **sound** in different languages, since they might be transmitted by voice, on the telephone, not

always by copying/pasting. Also, some systems might replace certain groups of printable characters, if they appear to be HTML-related!

Furthermore, our company is **already** able to to specify a desired EPP AuthInfo code at our preferred registrar (so this 'improvement' doesn't help us at all, and doesn't alleviate any of our concerns). Since this is one of the "improvements" that is used to justify removal of the Losing FOA, it's just completely inadequate.

Also, RFC 9154 notes in section 4.3 that "7. The registrar's interface for communicating the authorization information with the registrant MUST be over an authenticated and encrypted channel." While this is good, note that this requires that registrars NOT use SMS (not a secure channel). Also, email that is not sent over an encrypted port should be avoided (there are attacks that degrade security, that will force email to not use SSL, called "SMTP TLS downgrade attacks", see:

https://elie.net/blog/understanding-how-tls-downgrade-attacks-prevent-email-encryption/

https://powerdmarc.com/what-is-tls-downgrade-attack/

that registrars might not be aware of.)

Conceivably, section 4.3 could be redone to allow the registrant to be diverted to the registry, so that the registrar never has a copy of the TAC (if it's generated by the registry). The fact that the TAC goes through the registrar means that all kinds of attacks are possible (e.g. rogue code in the registrar that actually stores them somewhere, instead of keeping them as a transient value; even exotic attacks like Heartbleed or Row hammer:

https://heartbleed.com/

https://en.wikipedia.org/wiki/Row_hammer

can allow attackers to read memory that one would have thought as "protected".

A fundamental weakness of RFC 9154 is the omission of any audit trail capability (the word "audit" doesn't appear anywhere). Contrast that with our counterproposal in Section E, where the losing registrar **is allowed to log/save the PTID** (since it has absolutely no value to an attacker!).

Indeed, this just highlights that the working group has gone in a wrong direction, and has failed to systematically consider all the different security scenarios.

We love math (we own math.com after all) and quantitative analysis. But, RFC 9154 makes it seem to those who might be overly impressed with a few formulae and some technical jargon that it is some huge advance in security. It's nothing close to that.

Thus, both recommendations #7 and #8 do nothing to justify removal of the Losing FOA.

Rec #8: Significant change required: changing intent and wording

Identical answer to Rec #7.

See our prior discussion in our answers to Rec #5 and #6, especially: section J. AUTHINFO CODE / TAC IS RADIOACTIVE, TOXIC, DANGEROUS LIKE A KEY TO THE KINGDOM) is a high value target. By design, it should be eliminated (as our counterproposal in section E makes clear).

But, let's take this further. Rec #7 focuses on the complexity of the TAC, its length, etc. As we strongly argue in the other sections noted above, this complexity prevents a small class of attacks (i.e. brute force guessing of the TAC). It does nothing if the TAC itself is compromised between the time its generated and before it is used by the rightful registrant at the intended gaining registrar. That complexity does nothing to stop a whole slew of attack scenarios, as discussed above.

The problem is in the TAC itself being such a high value target that needs to be kept secret, not the length of the secret, etc. In contrast, our counterproposal in section E doesn't rely on the PTID (pending transfer ID) being complex -- it need only be unique for that domain. It can even be public, and a single character, and yet be **stronger** (due to the different process design) in terms of overall security.

Furthermore, we take issue with some of the guidance in Section 4.1 of the RFC 9154, which mentions printable ASCII character and also case insensitive characters. One might find it advisable to reduce the group of permitted characters even further, to eliminate those that look alike (e.g. zero (0) and capital "o" (O), and lowercase L (l) and capital "i" (I) and the digit "one" (1), depending on fonts). One might want to also pay attention to

how those letters/numbers/printable characters **sound** in different languages, since they might be transmitted by voice, on the telephone, not always by copying/pasting. Also, some systems might replace certain groups of printable characters, if they appear to be HTML-related!

Furthermore, my company is already able to to specify a desired EPP AuthInfo code at our preferred registrar (so this 'improvement' doesn't help us at all, and doesn't alleviate any of our concerns). Since this is one of the "improvements" that is used to justify removal of the Losing FOA, it's just completely inadequate.

Also, RFC 9154 notes in section 4.3 that "7. The registrar's interface for communicating the authorization information with the registrant MUST be over an authenticated and encrypted channel." While this is good, note that this requires that registrars NOT use SMS (not a secure channel). Also, email that is not sent over an encrypted port should be avoided (there are attacks that degrade security, that will force email to not use SSL, called "SMTP TLS downgrade attacks", see:

https://elie.net/blog/understanding-how-tls-downgrade-attacks-prevent-email-encryption/

https://powerdmarc.com/what-is-tls-downgrade-attack/

that registrars might not be aware of.)

Conceivably, section 4.3 could be redone to allow the registrant to be diverted to the registry, so that the registrar never has a copy of the TAC (if it's generated by the registry). The fact that the TAC goes through the registrar means that all kinds of attacks are possible (e.g. rogue code in the registrar that actually stores them somewhere, instead of keeping them as a transient value; even exotic attacks like Heartbleed or Row hammer:

https://heartbleed.com/

https://en.wikipedia.org/wiki/Row_hammer

can allow attackers to read memory that one would have thought as "protected".

A fundamental weakness of RFC 9154 is the omission of any audit trail capability (the word "audit" doesn't appear anywhere). Contrast that with our counterproposal in Section E, where the losing registrar **is allowed to log/save the PTID** (since it has absolutely no value to an attacker!).

Indeed, this just highlights that the working group has gone in a wrong direction, and has failed to systematically consider all the different security scenarios.

We love math (we own math.com after all) and quantitative analysis. But, RFC 9154 makes it seem to those who might be overly impressed with a few formulae and some technical jargon that it is some huge advance in security. It's nothing close to that.

Thus, both recommendations #7 and #8 do nothing to justify removal of the Losing FOA.

Rec #9: Significant change required: changing intent and wording

In addition to all the concerns about the TAC which we already expressed in our answers to Rec #7 and Rec #8 and above (i.e. TAC should be completely eliminated, in favour of our counterproposal in section E which is more secure by design, and uses a PTID which can be public!), in point #9.1 it doesn't contemplate the registry being the one to generate it (instead of the registrar).

Also, it seems, with all the emphasis on brute force attacks (which are not the only attack scenario), there should be lots of logging of failed requests at gaining registrars (who try to brute force the TAC of a domain elsewhere). That should be done and shared with the "targets" (i.e. the current registrar/registrant), and perhaps trigger enhanced security measures (although one would need to avoid a Denial-of-service impact, if someone intentionally blocks a legitimate transfer by generating lots of failed TAC submission requests at a rogue gaining registrar).

Rec #10: Support Recommendation intent with wording change

While we support the innocuous recommendation itself, one cannot claim that this is a "security improvement" in the rest of the document, to justify removal of the Losing FOA (i.e. Rec #10 has no improvement on security!). Go back to page 17 in the report where it says "The working group further concluded that if the TAC is managed in a more secure manner following Preliminary Recommendations 7-13, the risk of unauthorized transfer should be reduced." But, again, Rec #10 does nothing to bolster security.

Rec #11: Support Recommendation intent with wording change

While we support the innocuous recommendation itself (except to note again, like in other sections that the TAC should be entirely eliminated!), one cannot claim that this is a "security improvement" in the rest of the document, to justify removal of the Losing FOA (i.e. Rec #11 has no improvement on security!). Go back to page 17 in the report where it says "The working group further concluded that if the TAC is managed in a more secure manner following Preliminary Recommendations 7-13, the risk of unauthorized transfer should be reduced." But, again, Rec #11 does nothing to bolster security.

Furthermore, RFC 9154 says (in section 4.3):

"4. The authorization information MUST only be stored by the gaining registrar as a "transient" value in support of the transfer process.

5. The plain-text version of the authorization information MUST NOT be written to any logs by a registrar or the registry, nor otherwise recorded where it will persist beyond the transfer process."

But, according to Rec #11, the TAC is "one-time-use", and so if the registry operator has cleared the TAC, it would have had no loss of security if it had been logged for audit trail purposes by the gaining registrar. So, this didn't make sense.

Furthermore, as in an earlier section, it's unclear to us whether or not it's optimal for the losing registrar to be generating the TAC (as opposed to the registry).

Rec #12: Support Recommendation intent with wording change

While we support the innocuous recommendation itself (except to note again, like in other sections that the TAC should be entirely eliminated!), one cannot claim that this is a "security improvement" in the rest of the document, to justify removal of the Losing FOA (i.e. Rec #12 has no improvement on security!). Go back to page 17 in the report where it says "The working group further concluded that if the TAC is managed in a more secure manner following Preliminary Recommendations 7-13, the risk of unauthorized transfer should be reduced." But, again, Rec #12 does nothing to bolster security.

Indeed, if you wanted to actually bolster security and reduce the "attack surface", you'd actually **MANDATE** a **DELAY** in providing the TAC (just like the Google Advanced Protection and Authy examples that we mentioned in our answer to Rec #2). So, Rec #12 doesn't actually improve security -- it

just prevents a registrar from taking their sweet time to provide the TAC (like some registrars do intentionally, even after they've fully authenticated that a registrant wants to leave them, to annoy them). So, to the extent some people aren't keeping track of what's a security improvement and what isn't, this shouldn't "count" as a security improvement.

The better approach is to eliminate it through a different design, like the one in Section E above (BREAKTHROUGH PROPOSAL: GENERATE DOMAIN NAME TRANSFER TRANSACTION ID AT GAINING REGISTRAR TO INPUT AT LOSING REGISTRAR).

Rec #13: Support Recommendation intent with wording change

While we support the innocuous recommendation itself (except to note again, like in other sections that the TAC should be entirely eliminated!), one cannot claim that this is a "security improvement" in the rest of the document, to justify removal of the Losing FOA (i.e. Rec #13 has no improvement on security!). Go back to page 17 in the report where it says "The working group further concluded that if the TAC is managed in a more secure manner following Preliminary Recommendations 7-13, the risk of unauthorized transfer should be reduced." But, again, Rec #13 does nothing to bolster security.

Indeed, if you are a fan of the TAC (which we are not), how many people might get access to it over a period of **14 days**? This suggests that the attack surface is **much larger** than the working group might have implied (i.e. it's **not** just a case of a registrant with two browser windows open, one at the gaining registrar and one at the losing registrar, copying and pasting a TAC instantaneously, to thwart an attacker by keeping the time window "short"). Two weeks is a long time. Systems get hacked (even in the example we just made up, the registrant's system might have **already been** compromised, and the attacker was just waiting patiently on the network for the opportunity to come). And there are other scenarios, of course (where the TAC is shared with third parties like buyers and/or escrow, etc.).

Also, perhaps the language should be a MAXIMUM of 14 days (instead of "must be 14 days") in 13.1. While 13.2 allows the Registrar of Record (losing registrar) to cancel the TAC early, it seems some might want to have a shorter TTL when they provision it.

So, we reiterate, when one really examine things closely, one is left unimpressed as to the alleged "increase" of security that is claimed will occur from adoption of Recommendations #7-13.

Now that we've gone through Recommendations #7-13, we can safely conclude that the phrase "lipstick on a pig" certainly applies:

https://en.wikipedia.org/wiki/Lipstick_on_a_pig

(The phrase to put "lipstick on a pig" means making superficial or cosmetic changes to a product in a futile effort to disguise its fundamental failings.) The TAC is fundamentally broken, and Recommendations #7-13 are futile attempts to disguise its inherent insecurity. They certainly don't justify removal of the Losing FOA.

Question to the community: Who is best positioned to manage the standard 14-day TTL – the Registry or the Registrar, and why? Are there specific implications if the TTL is managed by the Losing Registrar?

Registry, obviously. That's where the (hash) of the TAC is used, so they'd be expected to know when it's been created, and can automatically invalidate it. They'd be able to check the timestamp (whenever it's been used at the gaining registrar) to see if it's beyond the TTL. And they can run a cron job (or other scheduled task) regularly (even daily) to check for TTLs on TACs that have been created, to see which need to be cancelled.

However, that being said, according to RFC 9154 (section 5.7) "7. If the transfer completes successfully, the registry automatically unsets the authorization information; **otherwise, the losing registrar unsets the authorization information when the TTL expires**; see Section 5.2."

So, if RFC 9154 isn't going to change from its current state, then it's the LOSING REGISTRAR that is supposed to unset the TAC.

In our opinion, the registry is in the better position to do so.

Again, this is all moot if we simply **get rid of the TAC completely**, as our counterproposal is Section E (BREAKTHROUGH PROPOSAL: GENERATE DOMAIN NAME TRANSFER TRANSACTION ID AT GAINING REGISTRAR TO INPUT AT LOSING REGISTRAR) would allow!

In other words, think outside the "TAC" box that the working group seems to have trapped itself within.

Rec #14: Support Recommendation as written

This is just terminology change, and has no impact on security.

Rec #15: Significant change required: changing intent and wording

We believe that communications/notices should also be sent to the tech contact, rather than just the RNH, for maximum opportunity of "actual notice" (as per prior section comments). [only the RNH would get a TAC, though; but notices can go to many]

Rec #16: Significant change required: changing intent and wording

First of all, we entirely reject the attempt to bring back the Expedited Transfer Reversal Process (ETRP) which is in the language directly above this recommendation (on page 31), and which we addressed above in relation to Rec #4.

Any restriction should be enforced by the registry, not registrar.

The language "initial registration date" is imprecise. It would seem to us that you mean "Creation Date" (a field that explicitly exists in WHOIS). Some folks (not us) consider a change of registrant to be the start of an "initial registration" (particularly in the UDRP policy debates).

The rationale listed in the rational regarding the UDRP is nonsensical, as the UDRP filing causes the name to be locked whenever it's filed, regardless of where the registrar is located.

Furthermore, this recommendation reveals enormous hypocrisy on the part of the registrars that dominate the working group. Registrars feel justified in adding delays in processes when there's a small chance they might lose money (e.g. fraudulent credit card payment). Let's suppose a 3% fraud rate, and a $20 domain fee, so an "expected" loss of 60 cents (less than a dollar!). So, 97% must be "inconvenienced" by 30 days, to guard against losing a maximum of $20.

But, if registrants want to keep an important safeguard in place (the losing FOA), because that small delay (it literally takes 20 minutes to get the ACK/NACK email at Tucows/OpenSRS) helps registrants avoid the harm of an unauthorized transfer, the "standard" changes, that this delay is somehow "unreasonable"?

To us, this seems like hypocrisy and a double standard. Delay in processes

are fine if it can save a registrar's bacon, but if similar (or even much shorter delays, since the Losing FOA isn't 30 days!) would save a registrant's bacon that delay is somehow **not** considered desirable!?!?!?!

Personally, we're indifferent as to the 30 days. but, the hypocrisy and logic compared to the reasoning with regards to elimination the Losing FOA should be pointed out!

Rec #17: Significant change required: changing intent and wording

(similar points to Rec #16)

Any restriction should be enforced by the registry, not registrar.

Furthermore, this recommendation reveals enormous hypocrisy on the part of the registrars that dominate the working group. Registrars feel justified in adding delays in processes when there's a small chance they might lose money (e.g. fraudulent credit card payment). Let's suppose a 3% fraud rate, and a $20 domain fee, so an "expected" loss of 60 cents (less than a dollar!). So, 97% must be "inconvenienced" by 30 days, to guard against losing a maximum of $20.

But, if registrants want to keep an important safeguard in place (the losing FOA), because that small delay (it literally takes 20 minutes to get the ACK/NACK email at Tucows/OpenSRS) helps registrants avoid the harm of an unauthorized transfer, the "standard" changes, that this delay is somehow "unreasonable"?

To us, this seems like hypocrisy and a double standard. Delay in processes are fine if it can save a registrar's bacon, but if similar (or even much shorter delays, since the Losing FOA isn't 30 days!) would save a registrant's bacon that delay is somehow **not** considered desirable!?!?!?!

Personally, we're indifferent as to the 30 days. but, the hypocrisy and logic compared to the reasoning with regards to elimination the Losing FOA should be pointed out!

Rec #18: Support Recommendation as written

Happy to support making language more precise.

Rec #19: Significant change required: changing intent and wording

We "sounded the alarm" to the community about this proposal in a blog

post:

https://freespeech.com/2022/08/01/double-red-alert-domain-registrars-seek-power-grab-to-deny-outgoing-transfers-of-legal-domains-they-dislike/

which apparently caused quite a stir in some circles. We note and support the Internet Commerce Associations's diplomatically-worded position. We would not be so diplomatic.

Frankly, we feel embarrassed for the working group that they have published such a one-sided proposal. It really demonstrates the imbalance that we discussed above in section B (WORKING GROUP SUFFERED FROM UNBALANCED AND UNREPRESENTATIVE PARTICIPATION). It's a huge power grab to attempt to deny a registrant the ability to change to a different registrar based on the disagreements over terms of use or anti-abuse policies (i.e. the changed text in I.A.3.7.1). It should be kept as-is, just for "evidence of fraud." This kind of policy change is more appropriately debated within an anti-abuse working group, rather than slipped into the transfers policy discussion.

We do not support GoDaddy's comment submission which attempts to add their enumerated list, either. It just belongs in a **different working group, focused on abuse issues** (not one that is supposed to be technical issues, which fewer people pay attention to). By doing less, the transfers working group will have more time to focus on the many other problems with this report (like the entire concept of the TAC, which we've highlighted repeatedly throughout our submission).

Potentially, this recommendation might even violate the ICANN Bylaws!

https://www.icann.org/resources/pages/governance/bylaws-en

which state in Section 1.1(c):

"(c) ICANN shall not regulate (i.e., impose rules and restrictions on) services that use the Internet's unique identifiers or the **content** that such services carry or provide, outside the express scope of Section 1.1(a)." [emphasis added]

ICANN expressly allowing a registrar to impose a restriction (denial of transfer of a domain) based on **content disagreements** seems to us to be very close (if not direct, perhaps indirectly) to violating the actual text, **if not the spirit of the text**, of that section (we're not lawyers, though).

Just to be clear, we're not here defending criminality. We want to ensure **due process** for registrants, especially given the typical power imbalance relative to registrars and/or registries.

As a note to ICANN staff, when you actually decide an order for going through the public comments, you might want to **start** with Rec #19, to just discard it first! We think if it's not discarded, it'll get a lot more attention from the public if it ever gets to the GNSO Council and/or the ICANN Board.

Rec #20: Significant change required: changing intent and wording

We do wish to mention here that the "general objection to all transfer requests received by the Registrar, either temporarily or indefinitely." can be made even **stronger** by our proposal in section I (TIMELOCK ACCESS TO TAC GENERATOR, AKA "VACATION MODE" or "LOCKDOWN MODE") of our submission.

Also, the "express objection to the transfer" will be ineffective if the Losing FOA is eliminated, and/or it's too late if the transfer has already completed before the registrant has the opportunity to learn of an unauthorized transfer. Thus, the Losing FOA must be retained (at least as an option!).

For the time limits in I.A.3.7.5 and I.A.3.7.6, it makes more sense to us for the registry to enforce the times, rather than the registrar.

Rec #21: Support Recommendation as written

Happy to support making language more precise.

Rec #22: Significant change required: changing intent and wording

"Silence" (i.e. no response) should be interpreted conservatively. So, in I.A.3.9.2, the default behaviour from a security point of view should be to deny a transfer, unless you have affirmative consent (i.e. if there's no "ACK", then the transfer should be denied, even if there's no "NACK"). That's how things once used to be, but then it got changed to where the transfer would automatically go through by default, if there was no response. This is the "UltraSecure" option mentioned above (section F. "THE BEST OF BOTH WORLDS" PROPOSAL TO RETAIN THE LOSING FOA ON AN OPT-IN BASIS BY REGISTRANTS) that registrants should be allowed to have the choice to opt-in to.

Small point, but in I.A.3.9.5, we've never liked the term "reseller" (although it's a defined term in the RAA). Some in the public (and UDRP complainants

in particular) might use the label to imply that a "reseller" must be registering domain names in order to resell them (rather than someone that simply has access to a registrar's advanced systems, to be able to access white-label platform for registration services, or a "pro" interface). That confusion over the meaning of "reseller" might negatively impact a registrant. So, it would be nice at some point to revisit the term "reseller", and perhaps replace it with "Value Added Integrator" or some other term that doesn't reference buying/selling domains themselves (as opposed to domain registration services). The "registration service provider" term is also fine.

Other Quick Comments (of course, there are also Sections A to J above!) :

On page 38, section 3.4.2, the term "Lock" with relation to Domain Name Locking for a UDRP should be made more precise. For greater certainty (since there is a presumption of innocence until proven guilty), registrant should be able to change nameservers during a UDRP (i.e. registrar can set clientTransferProhibited, but should NOT set clientUpdateProhibited). UDRP complainants had ample opportunity to collect evidence/screenshots before a UDRP was filed, and preventing nameservers to change might have a negative impact on a registrant. e.g. suppose a domain was hacked, and the nameservers were changed to those controlled by the "hacker". A UDRP filing shouldn't prevent a registrant from fixing those nameservers (to change them to what they were before they were hacked), all while keeping the domain at the current registrar. (registrars need to have better granularity for their locks, which they might not all do all the time, particularly when there's a UDRP).

Also, we're concerned about the "Next Steps" given that the "Undo" procedure (which we oppose, as discussed above, but might become a recommendation in Phase 1(b)) makes some of the recommendations interdependent. It should be possible to comment on whatever is finalized in Phase 1(a) again, and not just on something in Phase 1(b) if they interact with one another.

More generally, there's no attempt at an Impact Analysis in the report! There really needs to be a systematic review of potential attack scenarios, to be able to make it clear how **ineffective** the recommendations are in securing against them.

The Swim Lane seems to also incorrectly state "TAC Securely Stored" (by the registry). It's the HASH of the TAC that is securely stored (not the TAC itself that is securely stored).

If we were really going to "think outside the box" (like the XPRIZE suggestion), opening things up for radical changes (rather than incremental improvements), we'd suggest allowing registrants to opt-in to hardware-based authentication, coordinated centrally at the registry level. For example, we could enroll FIDO U2F keys (like Google Titan security keys, Yubico, etc.), and associate them with domain names (multiple hardware keys, obviously, for backups and redundancy). In the event of a transfer, the **same hardware key would have to be used at both the gaining and losing registrars**. Obviously, this would be a big change, and would need to be on an opt-in basis, but for those with more than 100 domains, or with valuable domains, we think there would be large adoption (more than 100,000 user accounts, representing tens of millions of domain names). Many people **already have these keys**, so it's just a matter of building an interface! Some registrars already support hardware keys, so they'd have better stats on adoption. But, if coordinated through the registry, it wouldn't require each registrar to build their own technology (it could be coordinated centrally).

L. ICANN PUBLIC COMMENT PERIODS ARE A SHAM. ALL PUBLIC COMMENT PERIODS SHOULD BE SUSPENDED UNTIL A FULL INVESTIGATION HAS OCCURRED

As a preliminary matter, we note with approval and fully support the Reconsideration Request 19-2 filed by Namecheap, Inc. regarding the .org contract renewal:

https://www.icann.org/resources/pages/reconsideration-19-2-namecheap-request-2019-07-22-en

where Namecheap wrote:

> The ICANN org will decide whether to accept or reject public comment, and will unliterally (sic) make its own decisions- even if that ignores the public benefit or almost unanimous feedback to the contrary, and is based upon conclusory statements not supported by evidence. **This shows that the public comment process is basically a sham**, and that ICANN org will do as it pleases in this and other matters. **It is a concern not only for the renewal of the .org** and other legacy TLD registry agreements being renewed in 2019, but an even greater concern for the upcoming renewal of the .com registry agreement- **as well as other vital policy issues under consideration by ICANN now and in the future**. [p. 12, emphasis added]

These are strong but thoughtful words from a highly respected company in the domain industry, whose views are shared by many. One of the synonyms for the word sham is fraud, and it's apparent now that a fraud has been perpetrated on the public, namely ICANN deceiving the public into believing that these comment periods were legitimate opportunities for meaningful input.

Their Reconsideration Request was (not surprisingly) denied by the Board, but it was escalated into an IRP that is still pending (ongoing for more than 2 years):

https://www.icann.org/resources/pages/irp-namecheap-v-icann-2020-03-03-en

The reconsideration request isn't strictly limited to the .org renewal, but directly calls into question the legitimacy of all of ICANN's public comment periods for all of the policy issues now and in the future. ICANN should not take their request lightly, but should instead call for a full public investigation with full opportunity for the ICANN community to weigh in on

this procedural matter which is at the core of ICANN itself. Until such an investigation has concluded, we call on ICANN to suspend all public comments periods, in order to ensure the process integrity of all policymaking.

Of course, given ICANN's comment process is a sham, this comment itself will likely be ignored, but we place it on the public record for posterity so that a higher authority will eventually hold ICANN accountable.

We would also like to note for the record that we submitted multiple related complaints to the ICANN Complaints Office regarding comment periods (including this latest comment period for the transfers policy report), which remain unresolved since they were initiated in April 2021:

https://www.icann.org/complaints-report

(see complaint number 00020671). They have a complete record of all the relevant materials/emails (which we won't include here, to save space/time). But, they appear to have simply vanished into a black hole, rather than being taken seriously by ICANN (with very long gaps in communications from the relevant staff, and no material updates). This reinforces the view that public comment periods are looked upon as a mere "box checking exercise", rather than a true opportunity for public input from impacted stakeholders.

"When a man sees his end... he wants to know there was some purpose to his life. How will the world speak my name in years to come? Will I be known as the philosopher? The warrior? The tyrant...? Or will I be the emperor who gave Rome back her true self? There was once a dream that was Rome. You could only whisper it. Anything more than a whisper and it would vanish... it was so fragile. And I fear that it will not survive the winter." - Marcus Aurelius, (from film "Gladiator")

"What we do in life...echoes in eternity." - Maximus Decimus Meridius (from film "Gladiator")

M. CONCLUSIONS

In conclusion, there is a lot wrong with this working group's report, too long to summarize briefly.

We wish to emphasize, though, that the TAC is very dangerous. It should be eliminated, by adopting our counterproposal in Section E (generate a transaction ID at the gaining registrar, to input at the losing registrar). This would be a huge improvement.

Also, the Losing FOA needs to be maintained, at least on an opt-in basis, as per our "Best of Both Worlds" proposal in section F.

Consideration should be given to inviting knowledgeable members of the ICANN community like ourselves to participate directly in the working group on behalf of domain registrants, to reduce the unbalanced participation that currently exists in the working group.

SSAC should also do a thorough review, in light of our own comments that highlighted the inconsistencies between their past advisories and the working group's recommendations. An XPRIZE-style competition would also be a way to bring in fresh ideas, and new solutions.