THINKST SCAPES

Q3 • 2025

https://thinkst.com/ts

Contents

Cover photo: Manyoni Game Reserve. *Image by Rob Hall*

Whale Trail Hike, De Hoop Nature Reserve, Western Cape. *Image by Bradley Jayanath (Thinkst)*

Introduction

Welcome to this edition of ThinkstScapes for Quarter 3, 2025! This issue focuses on content released, published or presented between the first of July and the end of September 2025.

It was a busy quarter, with the Hacker Summer Camp, HOPE, USENIX Security and more top-tier, large venues. Quarter 3 was the biggest of the year in terms of the amount of content reviewed, and there's some great work – so read on.

There was chatter online after DEF CON about some accepted talks that were potentially AI slop. We expect this will be a bigger problem down the line as review boards continue to struggle to judge a body of research from small, generically worded abstracts. Some conferences are setting explicit policies about the use of LLM-generated content in submissions, while others are trying to manage this manually. AI-generated content is becoming a problem across fields; the security community is not immune, we're certainly paying attention and hope you will too.

You can rest assured that ThinkstScapes write-ups are all lovingly human-crafted and edited by breathing fleshbags. The emdashes are (genuinely) all ours!

As a reminder: if you are aware of work we've missed, a blog post we should have seen or a conference we should have covered, we'd love to hear about it. Please send them to **ts@thinkst.com**!

In addition to ~1,300 blog posts, this quarter's content was drawn from talks and papers presented at the following conferences:

Conference name	Number of talks/papers
Black Hat USA	110
SEC-T	43
BSides Joburg	20
Security BSides Las Vegas	50
OrangeCon	28
fwd:cloudsec Europe	26
BSides Canberra	50
BSides Tallinn	18
WHY	37
USENIX Security	438
WOOT'25	18
Brucon	18
DEF CON 33	241
RomHack	7
HITCON	37
PancakesCon 6	24
44CON	21
VB2025 Berlin	67
ROOTCON 19	30
Blue Team Con	57
LABScon	38
STEELCON	18
SecTor	53
HOPE 16	91
SUMMERCON	11
Total	**1,551**

Themes covered in this issue

MICROSOFT-INDUCED SECURITY WOES

Microsoft's market power has a lot to do with its backward compatibility, but their Achilles heel is also tied to their long legacy tail. Works in this theme show how that tail keeps getting bitten. Starting with a critical cross-tenant token issue in Azure and phishing attacks from Entra ID's official page, then continuing with RPC security issues and conflicts with internal and external domain names, the hits keep coming.

LOGS ARE NOT ALWAYS AS THEY APPEAR

Works in this theme highlighted how logging, when performed at all, can be misled by edge cases or attacker sleight of hand. Look for spoofing cloud logs with cross-tenant VPC configurations, injecting into Windows event tracing, and low-level networking tricks to keep defenders confused.

AUTOBOTS ROLL OUT!

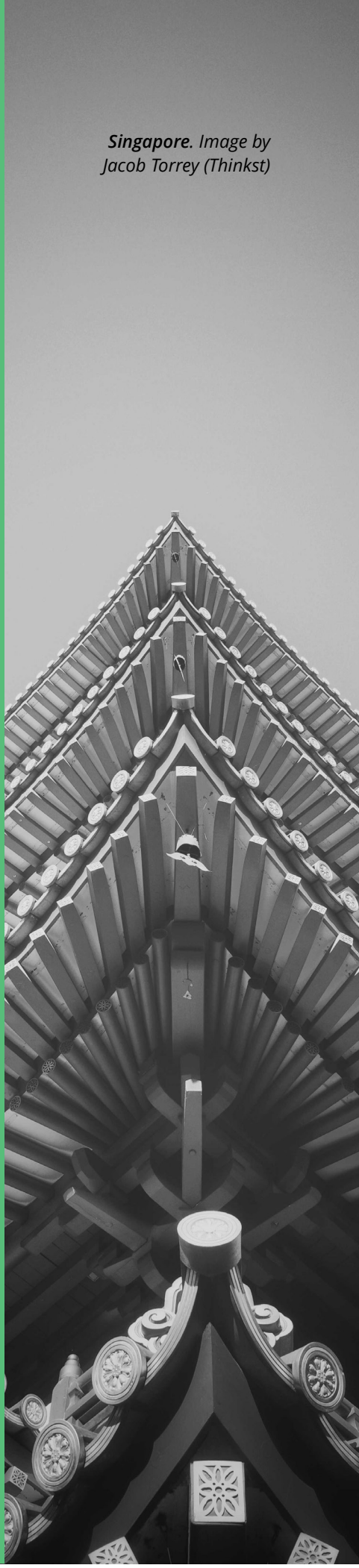
It's impossible to find a conference without a significant portion of LLM security talks. We feature four in this issue: a great retrospective from DARPA's AIxCC, data on how to combine models for better results, an analysis of how to manage false positives with LLMs, and using frontier models to automatically solve CAPTCHAs.

GOOD VIBRATIONS

This small theme looked at two research papers that use vibrations for digital security attacks. The first shows how modern computer mice can be used as bugs to eavesdrop on conversation, the second on how vibrations can change a computer's sense of time.

NIFTY SUNDRIES

As always, there was some stellar work we just had to feature that didn't fit our above themes. Look out for zero-knowledge digital identities, becoming a phone company, ML model attacks, low-level CPU attacks, a way to jam RAG systems, and breaking JavaScript's `Math.random()`.



Singapore. *Image by Jacob Torrey (Thinkst)*

Victoria Falls, Zimbabwe. Image by Jerry Maharaj (Thinkst).

Microsoft-induced security woes

- *One Token to rule them all - obtaining Global Admin in every Entra ID tenant via Actor tokens*
- *Turning Microsoft's Login Page into our Phishing Infrastructure*
- *You snooze you lose: RPC-Racer winning RPC endpoints against services*
- *Internal Domain Name Collision 2.0*

One Token to rule them all - obtaining Global Admin in every Entra ID tenant via Actor tokens

***Authors**: Dirk-jan Mollema*

Blog

This blog post covers a discovery by the researcher where impersonation tokens (Actor tokens) in Azure could be used to impersonate any user from any tenant, without any logging. These Actor tokens are used by some legacy Microsoft services (like SharePoint and Exchange) behind the scenes to synchronise hybrid environments. Because of this intra-MS service use case, there is little to no logging of their use – only tenant changes would be visible in audit logs. Microsoft also has not been transparent on exactly which services rely on these tokens.

The signed Actor JWT is inserted into an unsigned JWT when used to authenticate as that impersonated user (specified by the `nameid` field). However, the researcher determined that changing the tenant ID in the unsigned portions allowed for authentication as a user in that targeted tenant. There is still the need to recover the `nameid` for the victim user, but the post offers a variety of ways to do so, including brute-forcing since there are no logs or rate-limits.

TAKEAWAYS:

- **It's hard to overstate** how critical this issue was, and how fortunate the world is that it was found and reported.
- **The design of Azure and the legacy tail** of Microsoft applications combined weakens the cross-tenant protections offered by Azure. The legacy Azure AD Graph API needs to be ripped out. In-the-cloud features left in for some customers' backwards compatibility worsens security for all users.
- **This is not the first cross-tenant security issue** with Azure, and doubtfully will it be the last. Between the legacy features that refuse to fully go away and the lack of logs, it's tough to be optimistic that, without careful analysis of all authentication and authorisation flows, similar vulnerabilities will not continue to be discovered.

Figure 1.
A screenshot of the unsigned JWT actor token where the victim's tenant has been added using the attacker's signed token.

Turning Microsoft's Login Page into our Phishing Infrastructure

***Author**: Keanu Nys*

Slides

Video

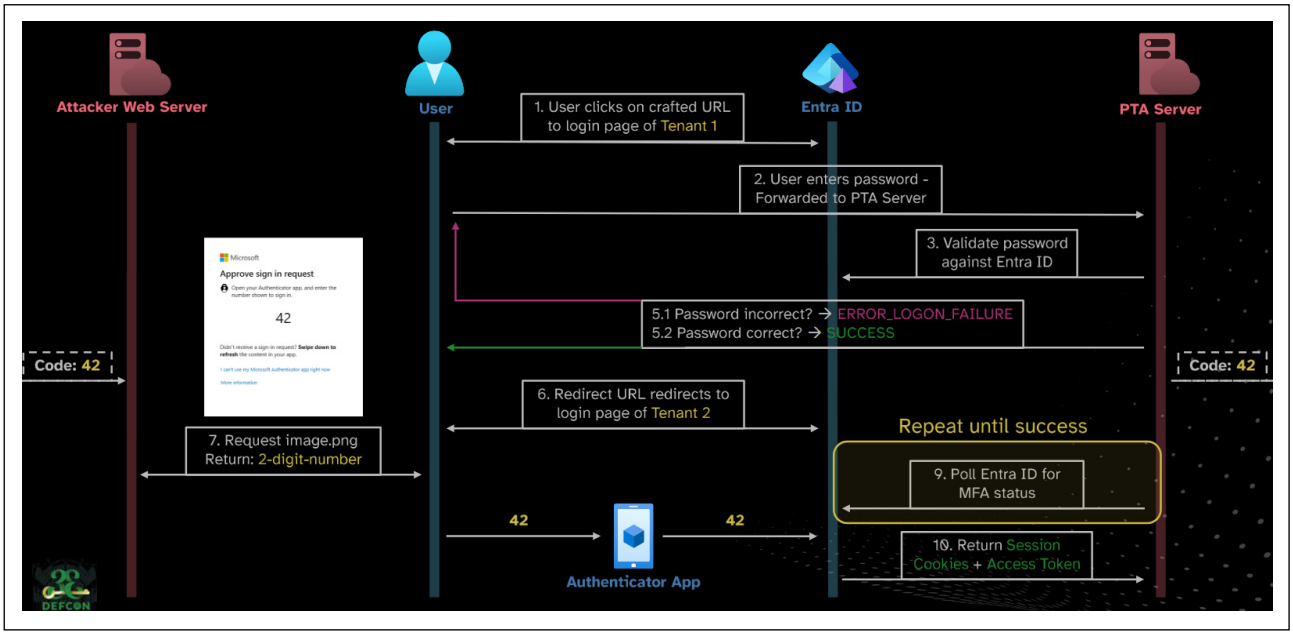
This talk showcased a number of ways to phish victims' Microsoft Entra ID credentials directly from the legitimate login.microsoftonline.com domain. Both user awareness training (to check the URL bar for suspicious domains) and email-scanning/firewalling new domains rely on being able to differentiate between valid and invalid domains to log in to. By crafting attacker-controlled Azure tenants, the researcher was able to build a host of techniques to have the domain in the phishing email and the URL bar (at least initially) be the real Microsoft-owned domain.

The initial techniques were methods of creating a tenant that would quickly redirect the victim to an attacker domain. While this bypassed email domain scanning and initial user investigation, firewall rules and a suspicious user would be able to see the new, malicious domain after the redirect. To improve on this, the researcher explored using a feature where login credentials are sent to an on-prem AD server for validation. By creating an on-prem AD server and installing malware on that server to log the credentials, if a victim is enticed to log in to the attacker's Entra ID tenant, the attacker can steal their credentials. To go further, the researcher created a custom domain that was close to the victim's tenant domain (e.g., micro-oft.com vs. microsoft.com), and using the custom branding feature created a font that rendered the '-' as an 's'. Finally, if MFA was deployed, the attacker server would redirect the victim to a second attacker tenant with a custom image loaded with the MFA code to enter. From the user's perspective the login flow is completely as expected and keeps the valid Microsoft domain in the URL bar, but the attacker is able to harvest credentials and bypass some forms of MFA.

TAKEAWAYS:

- **Phishing is a tough problem to solve as defenders**, due to humans playing a key role in the exploitation process. No software patch or bugfix is going to harden an organisation's users.
- **This talk highlights** that user awareness training and trying to stop phishing by scanning emails is a losing proposition. Moving to device-bound, phishing-resistant MFA is the only "silver bullet" to stop widespread identity theft – it's painful, yes, but essential.

Figure 2.
A sequence chart showing how to phish a user with MFA while keeping the user only on login.microsoftonline.com



You snooze you lose: RPC-Racer winning RPC endpoints against services

***Author**: Ron Ben Yizhak*

Slides

Code

Video

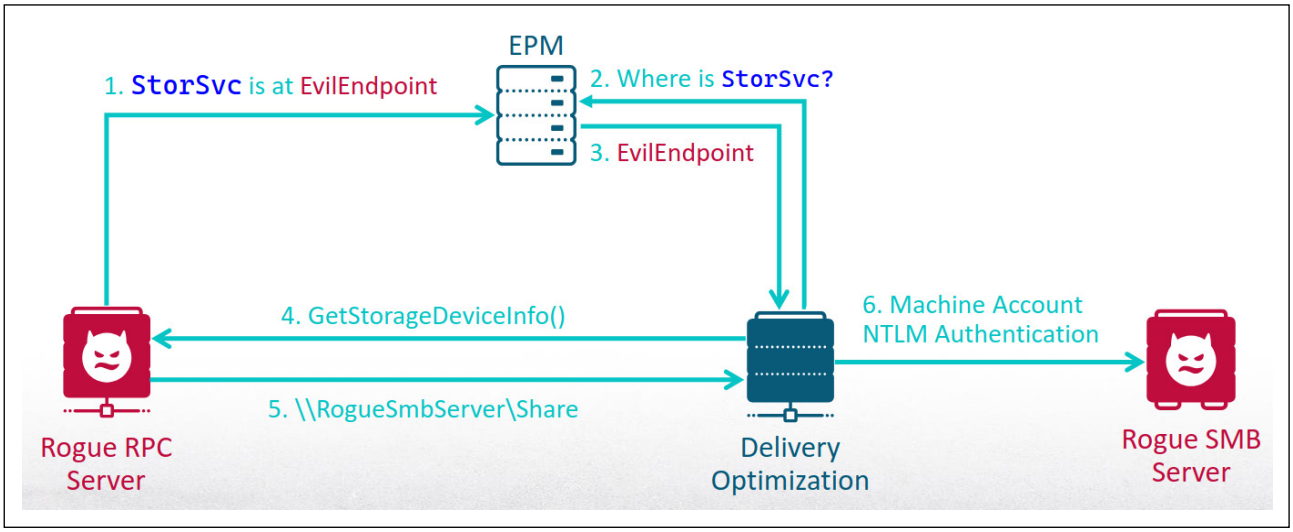
This research explored the security implications of registering a malicious RPC server as a well-known endpoint. Windows RPC is used internally to support inter-process communication, using a variety of transports. In order to locate the RPC service for a specific UUID or endpoint, the endpoint mapper service acts as a directory service for clients. There is no authentication needed to register an endpoint with the mapper daemon, so if a malicious service registers a well-known UUID before the legitimate service, client requests will be directed to the malicious service.

Since most RPC services are started early in the operating system boot process, the researcher explored which services were delayed in starting and registering their RPC endpoints. Next the researcher had to find ways to exploit a client as a malicious service. While many of the client processes were running with SYSTEM permissions, local exploitation attempts were challenged by checks on the service's permissions, or limited in the impact. However, by returning a UNC path to RPC queries for storage, the client would initiate an NTLM authentication (as the machine's AD account) to a server of the attacker's choosing. This ultimately led to privilege escalation, and depending on the circumstances, it could lead to a full AD domain controller compromise.

TAKEAWAYS:

- **While Microsoft has addressed this specific vulnerability** in StorSvc, the bug-class remains.
- **RPC research has seen a renaissance of late,** and without a seamless authentication scheme to prevent this race condition, more exploit chains will be discovered, just like with RPC coercion from a malicious client.

Figure 3.
A diagram showing how to escalate privileges by registering as the storage service RPC endpoint prior to the legitimate one. When the Delivery Optimisation service client connects, it can be passed a UNC path to coerce NTLM authentication to an attacker server.



Internal Domain Name Collision 2.0

***Author**: Philippe Caturegli*

Slides

Video

This research explored how conflicts between internal domains and the ever-expanding list of public gTLDs can impact security. While engaging on a red team, the researcher found that it was using an internal Active Directory domain of 'company.llc'. While this was reasonable in the past, '.llc' is now a gTLD that can have domains registered. When a corporate asset tries to connect to an internal service from an external network, it will resolve that domain using the public infrastructure and serve results that are controlled by the domain owner. In this case, NTLM hashes were sent over the public internet to a red team-controlled server.

The researcher wanted to explore whether this was a larger problem, so he looked for certificates that were found in public databases that were self-signed for domains that were unregistered. The gTLD '.ad' had a large number of hits: while clearly a good name for an AD domain, it's risky due to conflicts with Andorra's TLD. Dozens of large companies were found to have conflicts with, especially with the domain 'internal.ad', which the researcher was able to register.

Finally, the researcher found another class of domain conflicts: typos in NS records. MasterCard had a typo in one of their NS records that pointed to an Akamai CDN '.ne' domain instead of the correct '.net'. For less than $300, the researcher took over the '.ne' domain and immediately got traffic for the common typo, and not just for MasterCard.

TAKEAWAYS:

- **Changing an AD's internal domain** is nearly impossible. There are specific TLDs for internal or local traffic – use those to prevent a new gTLD from cropping up that will raise this issue.
- **If you're using a '.ad' TLD** for your network (or any public TLD), spend the money and register the public domain – the tiny cost for an extra domain name is well worth it.

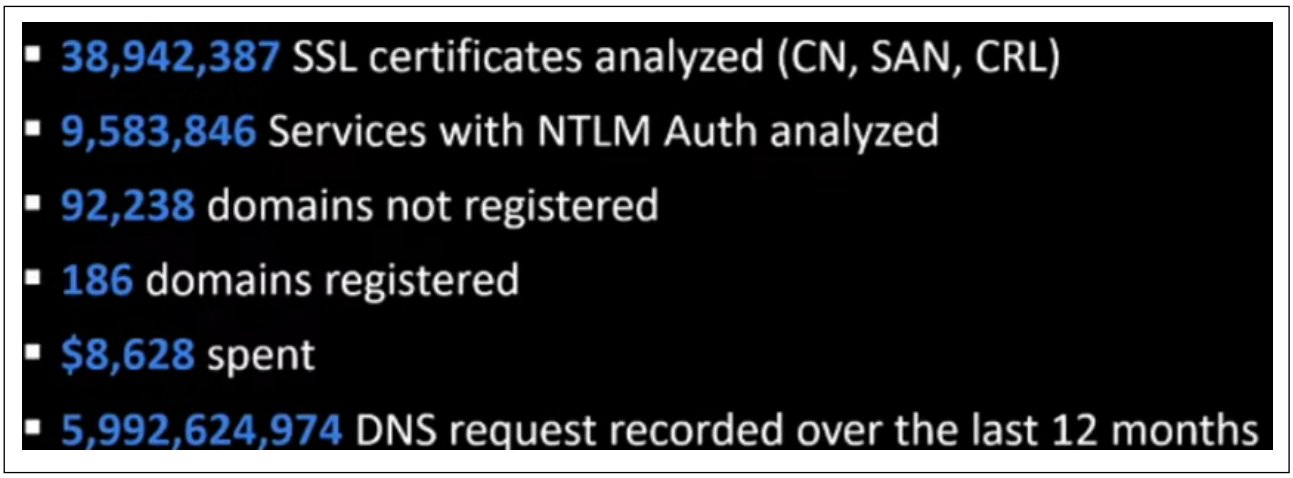


Figure 4.
A table of figures for the last 12 months of research into DNS conflicts.

Logs are not always as they appear

- ***Source IP Spoofing in Cloud Logs: A Hands-On Look Across AWS, Azure, and GCP***
- ***I'm in Your Logs Now, Deceiving Your Analysts and Blinding Your EDR***
- ***From Spoofing to Tunneling: New Red Team's Networking Techniques for Initial Access and Evasion***

Tugela Falls, Drakensberg. Image by Bradley Jayanath (Thinkst).

Source IP Spoofing in Cloud Logs: A Hands-On Look Across AWS, Azure, and GCP

***Author**: Eliav Livneh*

Video

Building off an incidental discovery from a few years ago, this research explored how to spoof the logged IP addresses in cloud logs. If an attacker were to acquire credentials for a victim's cloud tenant, the use of those credentials would show up in logs with the attacker's IP address – potentially raising suspicions. Until recently, if the attacker created a VPC in their own tenant, they could choose any IP address range for the VPC, and by using a VPC Endpoint, choose the source IP address logged by the victim.

The researcher then replicated this in both Azure and GCP, finding analogous approaches, though with some caveats. In Azure, the victim's tenant would have to have an approval (which would be logged) for the cross-tenant private link. In GCP, the IP address is simply marked as an internal IP, and no further data is provided.

In order to prevent or detect this type of spoofing, AWS now provides new fields in the logs for cross-tenant API calls, letting defenders alert on or even allow-list the other accounts allowed to use VPC Endpoints with their credentials. Azure users can audit the private endpoint approval process, or mark such approvals as a sensitive operation to limit the risk. GCP does not offer a specific defense, but alerting on IP addresses that are "gce-internal-ip" should offer a good start.

TAKEAWAYS:

- **Abstraction layers over cloud vendors** can only go so far when there are (attacker-induced) edge cases.
- **Operating in a true multi-cloud manner** is still a significant challenge, and requires careful inspection of how the abstraction is implemented in each cloud.

Figure 5.
A table summarising how an attacker can spoof logged IPs (and how defenders can prevent/detect the spoofing) across the three major cloud vendors.

	aws		
Attacker prerequisites (besides creds)	None	Approved private endpoint to victim tenant	None
Target service limitation	None	Only selected services, custom logging	None
IP spoofing control	Full	Full	"gce-internal-ip"
Indications of technique use	vpcEndpointId, vpcEndpointAccountId	None (yet)	None
How to detect	^ fields	Audit approved PEs	
How to prevent	aws:VpceAccount + aws:VpceOrgID condition keys	Prevent creation of new PEs	(* VPC-SCs for specific hardening)

I’m in Your Logs Now, Deceiving Your Analysts and Blinding Your EDR

***Author**: Olaf Hartong*

Slides

Code

This research explored manipulating Microsoft’s event tracing for Windows (ETW) subsystem used by many EDRs for detection. To improve the robustness of the operating system (especially after the CrowdStrike fiasco), there has been a push towards user space for more system functionality. ETW provides a single API to access events from both user space and kernel providers. By subscribing to these event streams, EDR solutions can gain visibility into sensitive events without themselves needing kernel level access.

The core primitive this research highlights is that multiple processes can generate user space events for the same GUID. Events stemming from a specific GUID are treated as coming from that event log, allowing the researcher to inject events that appear to come from a core Windows service. By enumerating the event streams consumed by EDRs, the researcher could inject events, trigger alerts for phantom detections, and overflow the EDR caps.

In order to minimise network consumption, each EDR product caps the amount of traffic sent to the cloud-based detection system per event type. By inserting enough fake events of a specific type, the EDR would not see (or alert on) a real event within the capped window. Finally, it is possible to overflow the ETW buffers and prevent any logging at all for specific streams until a reboot.

TAKEAWAYS:

- **The majority of EDR evasion** has been focused on avoiding generating events, whereas this work showed the attacker benefits of injecting fake ones.
- **EDR vendors walk a fine line** in capturing the telemetry to capture malicious events while also minimising performance impact and network bandwidth consumption. Expect to see more abuse of event injection across the spectrum of defensive tooling as this becomes popular.
- **With event roll-up and capping,** injecting enough obvious false positives to cover a real attack is a sound approach. Security product vendors must ensure that their logic doesn’t group similar events too broadly, or defenders will lose visibility when it matters.

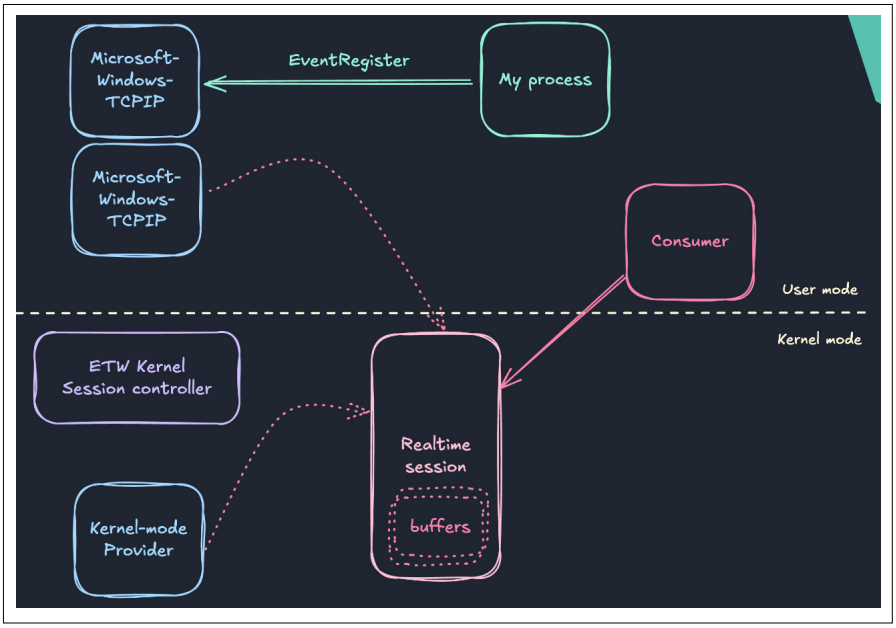


Figure 6. *A diagram showing how a process can register itself as an ETW event provider with the same GUID and identity as an existing provider.*

From Spoofing to Tunneling: New Red Team's Networking Techniques for Initial Access and Evasion

Author: *Shu-Hao Tung*

Slides

Paper

Video

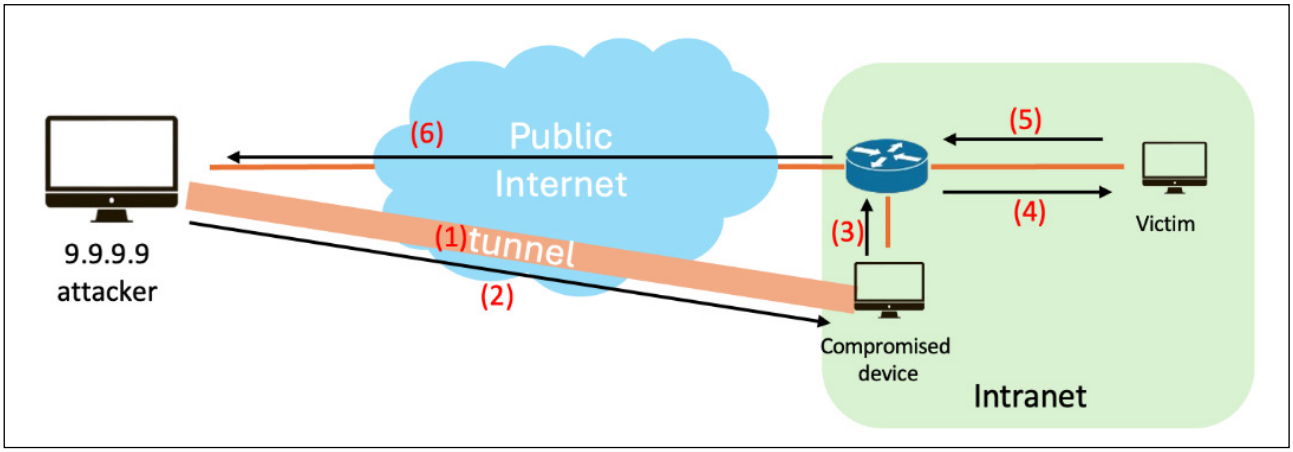
This research explores the security implications of using spoofed source IP addresses in a variety of network configurations. While many IPs prevent packets from spoofed source addresses, there is often no ISP blocking within a LAN/subnet, or encapsulated in valid, non-spoofed packets. Internal to a LAN (depending on subnetting and segmentation), a compromised host can set the spoofed source IP to the attacker's external IP, frustrating detection engineering and IR. In logs, it will look like the internal host was in communication with only the attacker's external IP, increasing the difficulty of identifying the attacker's initial foothold.

The second class of presented techniques are those targeting GRE tunnels. GRE tunnels provide organisations with multiple sites to easily bridge internal LANs. When a packet destined for another site's subnet reaches the router, the router encapsulates the packet with a header to the external IP for the other site's router. That router strips off the encapsulation header and sends the packet on. Many GRE tunnels operate without encryption, and use static IPs as the method to authenticate the traffic from the tunnel. By sending encapsulated ICMP Ping packets encapsulated to the target router with an internal source IP of the attacker's external IP, and spoofing the encapsulating header, the researcher can scan a router for configured tunnels and then access internal LAN hosts.

TAKEAWAYS:

- **While the security considerations** of basic NATs are well-understood by most, edge cases can be dangerous with source spoofing.
- **Most blue teams would struggle** to trace misbehaviours coming from a spoofed external IP – plan now for how to get visibility into layer two for when it's needed to root cause a detection.
- **GRE tunnels may have been reasonably robust** prior to the era of IPv4-wide scanning and Shodan-style OSINT repositories. If a source IP is used as a form of authentication, expect for a motivated attacker to bypass that protection in due time.
- **Any non-authenticated and encrypted tunnels** should be replaced to prevent both these types of attacks as well as routing internal network traffic across untrustworthy links.

Figure 7.
A diagram showing how a compromised host can hide its lateral movement to the victim machine by spoofing the traffic's source IP to that of the attacker's external IP.



Autobots roll out!

- *Automating software security with LLMs*
- *Agents Built From Alloys*
- *AI Agents for Offsec with Zero False Positives*
- *Are CAPTCHAs Still Bot-hard? Generalized Visual CAPTCHA Solving with Agentic Vision Language Model*

Lake Tanganyika, Zambia. Image by Daniel de Villiers (Thinkst)

Automating software security with LLMs

***Author**: Tyler Nighswander*

Website

Code

Video

This talk provided a review on DARPA's AIxCC challenge to use LLMs to automate software vulnerability discovery and repair. The presentation is by the team lead for the third place team in the final competition, and goes beyond the architecture and results of the implementation. Working backwards from a correct patch, the entire problem of vulnerability discovery and patching is decomposed, and various approaches are compared.

By tracking the costs of each step, it could be optimised for in different ways. For example, in some cases it was cheaper to use a smaller model multiple times than a better model once. The researcher shares insights in the balance between the upfront costs of culling false positives earlier or the increased costs of trying to build a proof-of-vulnerability for a bug that doesn't exist.

The talk also includes a walkthrough of the tool finding a true zero-day in an open-source target as opposed to the inserted vulnerabilities from the AIxCC competition. By exploring regions of code to stimulate, writing input generators, and tasking debuggers, the LLM was able to find a complex payload that triggered the vulnerability, making it suitable for testing candidate patches.

TAKEAWAYS:

- **Beyond the details of the architecture and implementation** for bug finding, this talk shares many lessons for how to decompose a large problem for automation with LLMs.
- **Between delegating sub-tasks** to other LLM tools or deterministic processes, and using an LLM to generate Python code as opposed to generating inputs directly, there's a lot to take away for anyone architecting an LLM-supported system.
- **The intuition that providing the model** with too many options can hamper performance is interesting. Oftentimes LLMs are anthropomorphised into entities that can do what humans can but at much larger scales. The data shows that results from LLMs are much better when the context is kept smaller, and fewer options are available to choose between.

Figure 8.
A hexdump showing a fully LLM-discovered and developed proof-of-vulnerability in the FreeRDP project.

Agents Built From Alloys

***Author**: Albert Ziegler*

Blog

Dataset

This blog post explored a method for combining (alloying) two LLMs and how alloying improved the success rate in bug-finding tasks. As new models are released, the research team uses their benchmarking challenge suite to measure performance. In reviewing the evaluation data, the researchers identified that there was a relatively clear delineation between the challenges that one model would solve better than another. While both Sonnet 4.0 and Gemini 2.5 perform well, there are some challenges where one model far outperforms the other.

This led to the notion of alloying the two models together to attempt to get the diversity of approach from both models. For a given task, the models are switched between, but with the context, inputs and outputs from both models provided. This does not change the amount of model invocations needed, and each model is prompted such that the past context is not associated with a specific model. Each model is unaware of the alloying process, but benefits from the increased diversity of output. The improvement is better than the sum of the parts; more challenges are solved by alloying than by running the test suite with both models individually.

Finally, the blog post notes that alloying works best with two capable models that have different strengths and weaknesses. Adding a weaker model will pull down performance, and/or combining two similar models will yield little improvement.

TAKEAWAYS:

- **Ensembling models,** or employing a mixture of experts has been a clear path to boost (no pun intended) the performance of learners for decades. However, the presented alloying technique can provide the benefits of such techniques for consumers of closed-source models without significantly increasing costs. With a common interface language (any language that the models can understand), alloying should provide benefits for a wide-variety of tasks.
- **The success of the alloying technique** is built on the past evaluation effort to measure the varying levels of success between models and see which models would be the most complementary. Getting great results from LLMs still requires significant research and effort.

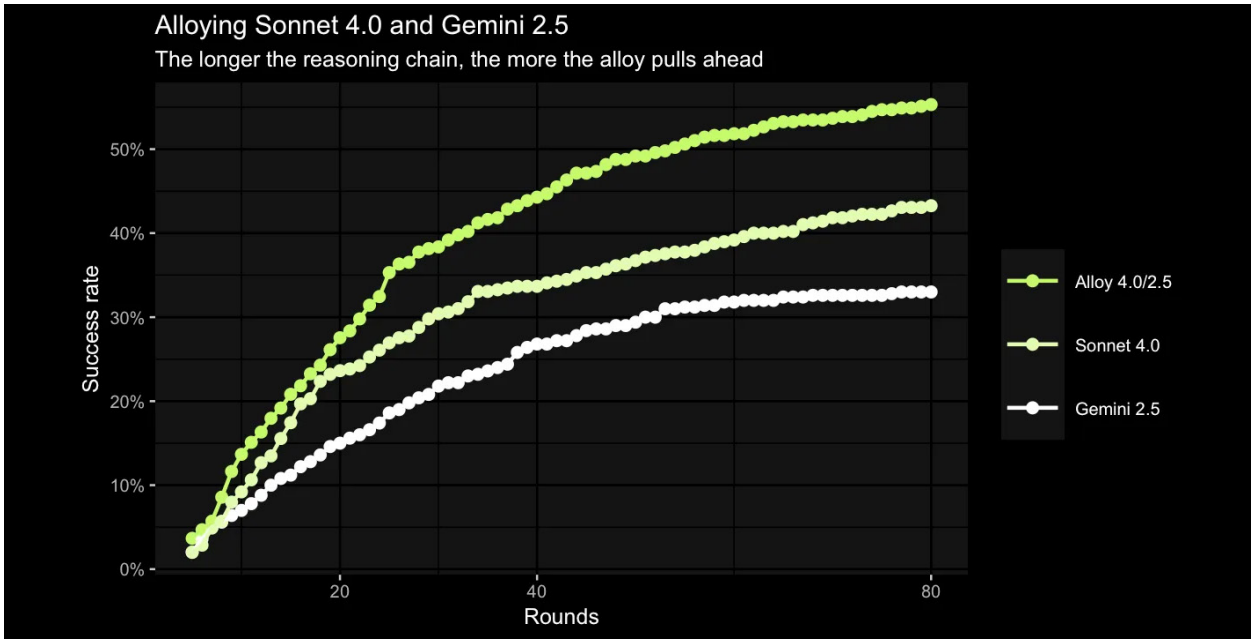


Figure 9. *A chart showing XBOW's success rate at solving security vulnerability discovery tasks over multiple rounds with two frontier models and an "alloy" of both.*

AI Agents for Offsec with Zero False Positives

***Author**: Brendan Dolan-Gavitt*

Slides

This talk by a researcher at XBOW detailed some of their thinking on how to use LLMs as bug-finding agents. XBOW's LLM agent system was HackerOne's top bug reporter for a time. At scale, even a modest false positive rate becomes noisy and can sour receiving organisations of hallucinated bug reports. The talk focused on how to structure the environment to eliminate false positives (at the cost of some true positives) depending on the target and level of manual effort available.

By putting a unique (i.e., a UUID) flag into the environment in a location that would require the LLM agent to find and exploit the bug of a certain class, the report is only trustworthy if the flag matches. For example, an SSRF bug would be tested by a flag put on a web server on an internal server, and the SSRF report is only considered valid if the flag was successfully recovered. By putting flags into different places in the system, it is possible to reduce false positives while still discovering hundreds of true positive vulnerabilities in real software systems.

TAKEAWAYS:

- **When comparing XBOW's approach** to less-mature entities aiming at the same target, it is interesting to see how carefully scaling is considered. At scale, false positives are what will cause LLM bug-finding systems to be written off as useless slop – thus prioritising low FPs and accepting some misses.
- **The results of a validation-first approach** speak for themselves: hundreds of valid, accepted bugs in production software with minimal manual effort. Expect to see this approach duplicated and scaled up considerably in the near-term.
- **As early fuzzing tools gained popularity,** they too opened a massive seam of vulnerabilities. It will be interesting to see if this vulnerability-finding technique will continue to find bugs after the initial burst, or if the classes it is best-suited for are quickly tapped out.

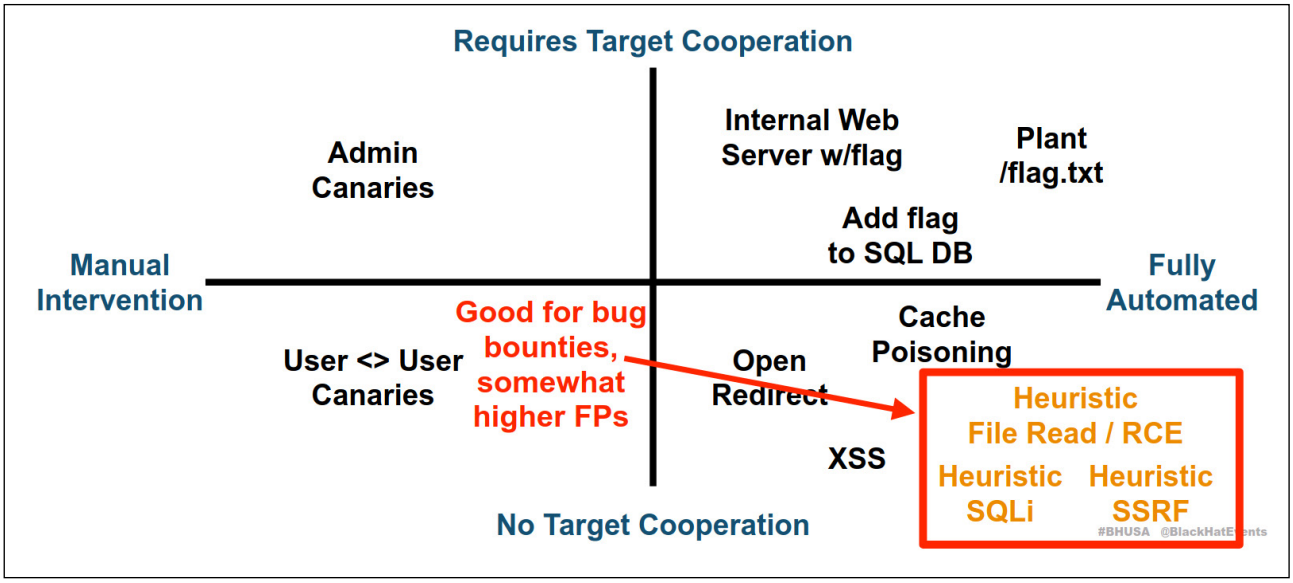


***Figure 10.** A quadrant chart showing different techniques for validating an LLM's ability to exploit a system.*

Are CAPTCHAs Still Bot-hard? Generalized Visual CAPTCHA Solving with Agentic Vision Language Model

***Authors**: Xiwen Teoh, Yun Lin, Siqi Li, Ruofan Liu, Avi Sollomoni, Yaniv Harel, and Jin Song Dong*

Website

Paper

Code

This work explored using modern multi-modal LLMs to generically solve CAPTCHAs. As anyone who's tried to access a modern website knows, CAPTCHAs aim to prevent bots or automation from accessing the resources they protect. These tests have changed over time, both with more vendors offering CAPTCHAs-as-a-service, but also in response to the improvements in computer vision algorithms.

There are a number of existing CAPTCHA solvers that have been optimised and trained on specific types of captchas, but this work, Halligan, evaluated how a LLM could solve CAPTCHAs in general. Using the visual input capability of many recent LLMs, the CAPTCHA was identified from the web view. The model would then try to understand the objective to solve, and then attempt to solve the CAPTCHA (by generating code to interact with the browser). Overall, Halligan was able to automatically solve over 60% of CAPTCHAs drawn from 26 different services (if three retries are allowed, this results in a 95+% solve-rate). 3D spatial reasoning was the worst-performing challenge type, but was also found less frequently in the wild. Halligan was then wired into a human-powered CAPTCHA solving service as a worker, and was able to get credit for completing over 70% of the 3000 CAPTCHAs it was assigned. The median time to solve a CAPTCHA was just over 20 seconds, costing $0.024 in OpenAI API costs.

TAKEAWAYS:

- **LLMs have been extensively analysed** for their ability to generate content and how that content can impact security in various ways. This work looks at how powerful multi-modal models can understand visual elements in a way that allows them to solve the very challenges designed to differentiate humans from bots. This is another modality where LLMs are shaking up security.
- **While a few candidates** were offered at the end of the paper for new CAPTCHA models, the widespread defeat of existing CAPTCHAs must spur innovation in this area. If your organisation is protected by CAPTCHAs currently, it's worth looking for alternatives in this space.
- **While many computer vision-based security models** have prioritised speed (such as Chrome's phishing detection ML), there are a number of defensive use cases where multi-modal LLMs could offer security benefits. LLMs don't get decision fatigue, and can be uniformly trained en masse to act in a supporting role for helping protect end users. As LLMs become smaller and more efficient, and hardware more powerful, expect to see a new market for on-device security assistants.

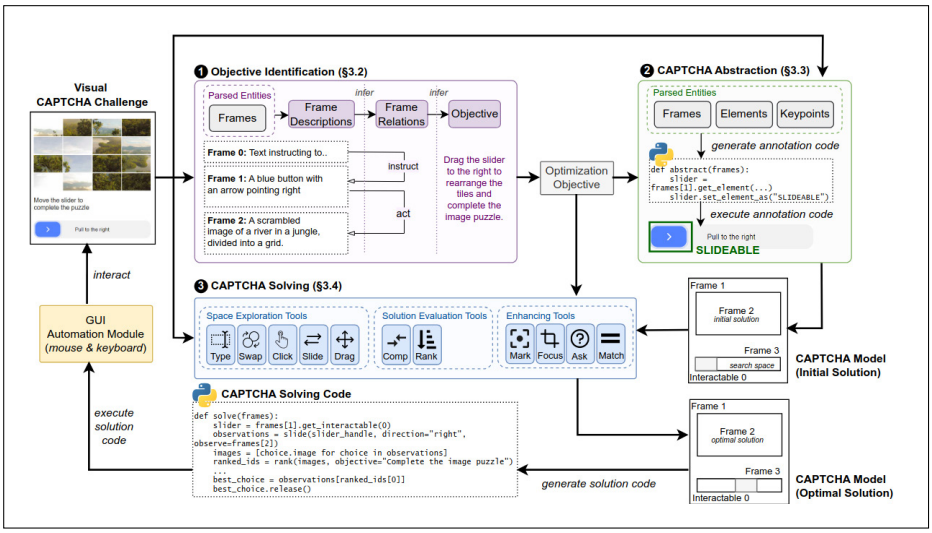


***Figure 11.** A flowchart showing how the Halligan tool works to understand the CAPTCHA's objective, abstract the specific graphic elements, and to build code to solve the CAPTCHA.*

Good vibrations

- *Invisible Ears at Your Fingertips: Acoustic Eavesdropping via Mouse Sensors*
- *TimeTravel: Real-time Timing Drift Attack on System Time Using Acoustic Waves*

Omar Ali Saifuddien Mosque, Bandar Seri Begawan, Brunei.
Image by Jacob Torrey (Thinkst)

Invisible Ears at Your Fingertips: Acoustic Eavesdropping via Mouse Sensors

***Authors**: Mohamad Habib Fakih, Rahul Dharmaji, Youssef Mahmoud, Halima Bouzidi, and Mohammad Abdullah Al Faruque*

Website

Paper

This research explored how to remotely recover speech spoken in proximity to an optical mouse. The last decade has seen optical mice improve both their DPI sensitivity and sampling rate – originally to cater to gamers and artists. Modern operating systems expose the high-precision mouse positioning data to applications without the need for special permissions. Games, digital art software and CAD applications make use of these APIs for more precise mouse capture – requiring administrative or kernel permissions would be too cumbersome.

The minute vibrations induced in the optical sensor from sound waves cause minor fluctuations in the position reports. With filtering and neural-network-based processing, the researchers were able to recover 41% of the speech through speech recognition algorithms. When only selecting for spoken digits (such as a PIN), that accuracy increased to over 61%. Additionally, the research explored how the surface the mouse was placed on impacted recovery accuracy – rough and cushioned materials provided more privacy than smoother surfaces. Mouse pads significantly decreased the attack performance, limiting the real-world applicability of the attack.

TAKEAWAYS:

- **The cost-capability curve** of many electronics has exponentially increased with little-to-no mainstream awareness. That a mouse today costs half as much as one from less than a decade ago while offering twice the DPI and 800% the sampling rate is impressive. With this increased amount of data precision and sampling rate, it makes sense that more side-channels will be discovered, or made feasible with the advancements in ML.
- **Disabling or blocking microphones** is commonplace in sensitive environments, and this work shows that, for a motivated attacker, microphones can be synthesised by other sensor sources. Past research has found that gyroscopes in mobile phones can also be used to recover audio. This advancement puts desktops in secure office spaces at risk.

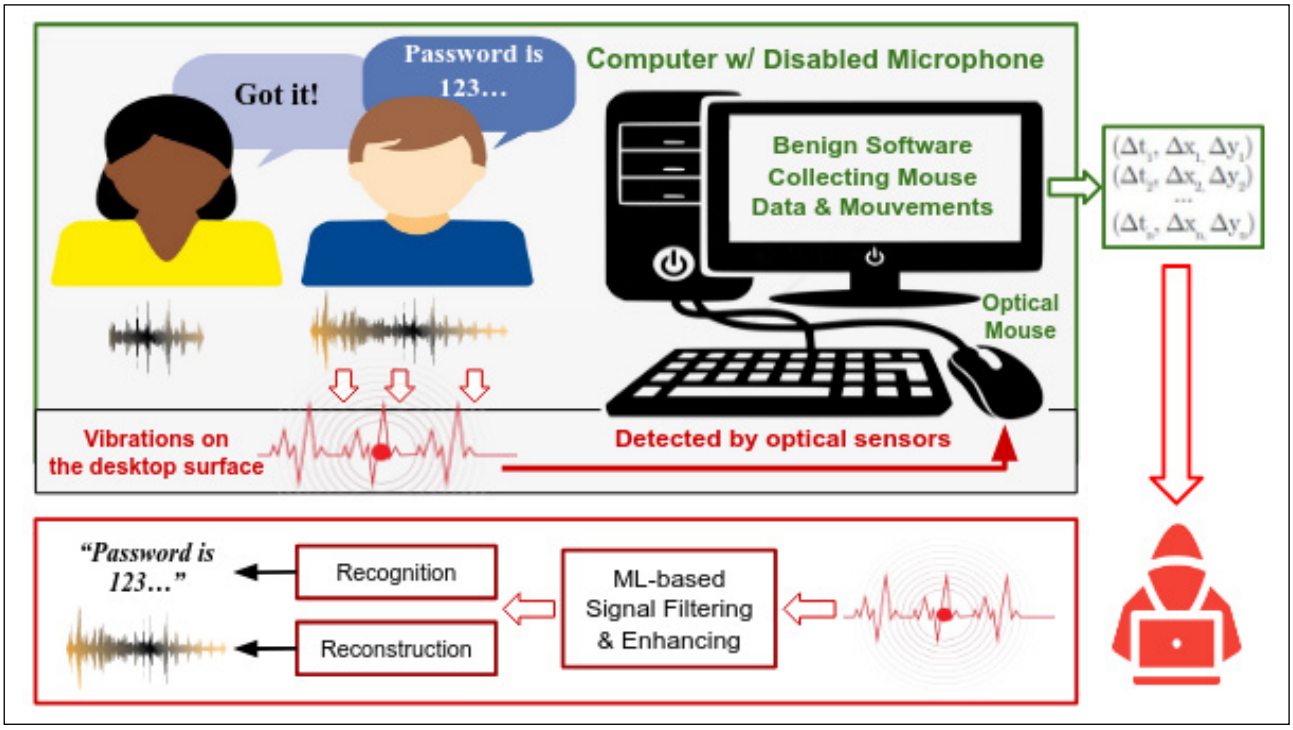


Figure 12. *A high-level diagram showing a potential concept of operations for using a mouse to eavesdrop.*

TimeTravel: Real-time Timing Drift Attack on System Time Using Acoustic Waves

***Authors**: Jianshuo Liu, Hong Li, Haining Wang, Mengjie Sun, Hui Wen, Jinfa Wang, and Limin Sun*

Paper

Timing is a crucial component of many digital processes, from cryptographic operations to accurate sensing. This work explored using acoustic waves to attack and alter the real-time clock (RTC) component integrated into digital devices. By using a magnetic probe placed in close (1-7.5 cm) proximity to the target device to sense the clock's oscillation, acoustic vibrations can be injected via a transducer stuck onto the same surface (20+ cm away). These vibrations can either be coordinated to increase the number of detected edges within the RTC's logic, or work to cancel out the crystal's own vibrations. These finely-tuned vibrations can either increase or decrease the number of recorded "ticks", changing the device's perception of time.

The researchers were able to measure the impacts in consumer devices with varying distances from the transducer, different materials for the surfaces the target device was placed on, and intensity of the acoustic vibrations. The power levels needed to impact the devices were within the normal background noise levels and were robust to typical other vibrations present, such as a desktop fan.

Across the tested devices, the attack was successful at skewing the time +/- 25s in a 30s attack window in more than 80% of the trials. Softer materials (such as firm rubber) required more power than hard surfaces, but the attack was still feasible.

TAKEAWAYS:

- **While it is unlikely** that many readers will ever be impacted by this attack, it's important to remember that physical phenomena underlie the digital logic in all devices.
- **Timing plays a crucial role** in cryptographic protocols: being able to impact the apparent time from a stand-off distance could have implications for deployed sensors or IoT devices.

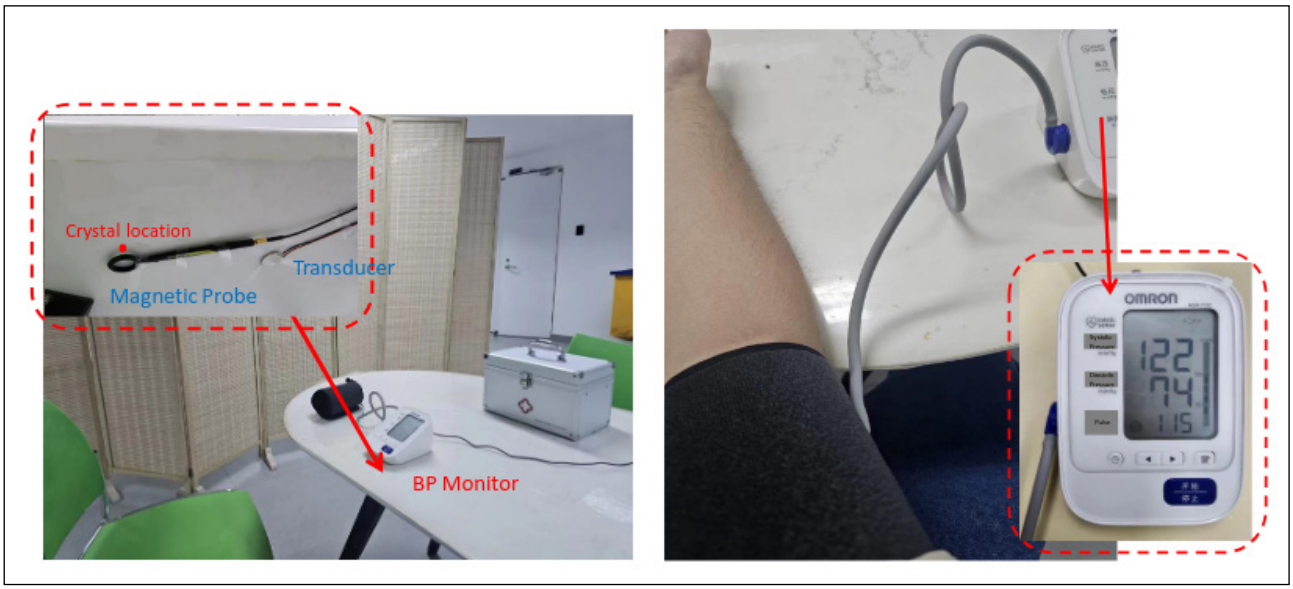


Figure 13. *A photo of a conceptual deployment of the TimeTravel attack where the probe and transducer is affixed to the underside of a table and can alter the blood pressure readings by skewing the RTC performance.*

Red Rock Canyon, Las Vegas, USA. Image by Marco Slaviero (Thinkst)

Nifty sundries

- *Crescent library brings privacy to digital identity systems*
- *Journey to the center of the PSTN: How I became a phone company, and how you can too*
- *Safe Harbor or Hostile Waters: Unveiling the Hidden Perils of the TorchScript Engine in PyTorch*
- *Ghosts in the Machine Check – Conjuring Hardware Failures for Cross-ring Privilege Escalation*
- *Machine Against the RAG: Jamming Retrieval-Augmented Generation with Blocker Documents*
- *Inverting the Xorshift128+ random number generator*

Crescent library brings privacy to digital identity systems

***Authors**: Christian Paquin, Guru-Vamsi Policharla, and Greg Zaverucha*

Blog

Paper

Code

This blog post highlighted an open-source Microsoft library, Crescent, that offers the ability to use existing web token standards in an unlinkable manner. Many services require proof from a third-party (such as proof of age, identity or employment), where users are forced to either avoid using the service, or offer sensitive personal information. Using an existing JWT or other cryptographic identity token, Crescent can provide proof that the user has such a token without linking it to the user's personal information. The zero-knowledge proof system also allows for the user to determine what information is provided, and how it can be further abstracted. For example, if a site needs to verify age, rather than revealing the user's date of birth, it can simply provide a proof that the user is over the required age.

While the preparation of the zero-knowledge proof can take on the order of minutes, this step only needs to happen once per credential. The verification process takes 200 ms at most. This overhead is perfectly reasonable for a site to offer privacy-preserving verification based on multiple types of original tokens, including mobile wallet tokens (such as digital driver's licences).

TAKEAWAYS:

- **Blocking things online** has become the new hotness for the "protect the children" crowd. It's created a patchwork of regulations for countries and states that are doing little, other than driving VPN adoption. Adult content, social media, and private spaces (such as gender-restricted discussion forums) all have some restrictions being placed on them – requiring privacy-leaking verification. Users are rightfully distrustful of uploading photos or videos of themselves and their government-issued IDs to a service that may leak them to the public; tools like Crescent offer a viable solution.
- **With more widespread deployment** of unlinkable credentials, online spaces can be more carefully curated. Social media services could ensure that users were real people with a government-issued ID without forcing users to reveal their identities.
- **Thanks to the backward compatibility** of Crescent, it's not too far-fetched to imagine a time where digital identities are tied to real identities in a privacy-preserving manner. This could improve security by tying network traffic to users, but also increase damages from stolen credentials.

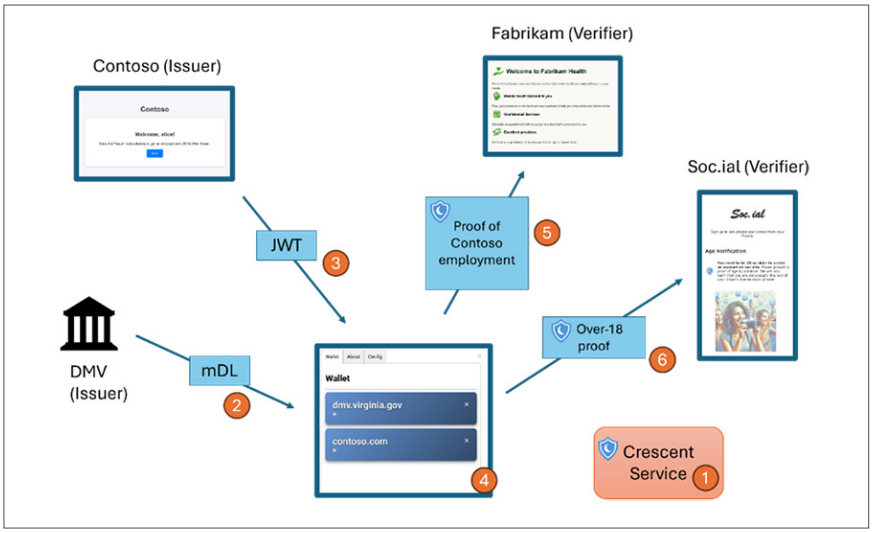


Figure 14. *A diagram of how a user can use Crescent to provide unlinked proofs derived from existing cryptographic tokens to verifiers with specific, selective attributes.*

Journey to the center of the PSTN: How I became a phone company, and how you can too

***Author**: Enzo Damato*

Slides

Video

This talk explored the current (US) public switched telephone network, or PSTN. Stemming from the breakup of AT&T to the current mish-mash of providers and interconnects, making a call is not a simple process. Complex and inconsistent regulations have allowed for bad actors to thrive. Highlighted are both access stimulation (generating revenue from driving fake calls) and caller ID spoofing. The FCC has tried to regulate both using a mix of market rules (such as ruling certain types of calls ineligible for payment) and technical means (caller ID verification) with mixed success.

The researcher concludes with the procedure for registering as a telecom operator and getting a bank of phone numbers and peering arrangements. This blueprint includes lessons learned and timelines for each step along the way (each measured in months).

TAKEAWAYS:

- **In an era where zero-trust is all the rage**, it's fascinating to glimpse at a security-relevant community that is primarily all based on trust (and slow bureaucratic processes). Even with regulators trying to cut down on caller ID spoofing spam calls and SIM swapping, the incentives in this industry reward moving slowly, and thus getting paid by bad actors.
- **Don't expect the state of PSTN security** to improve unless there is a bad actor who draws enough ire from policy-makers (e.g., by robocalling people with fake election information).
- **It's hard to quantify the value** of exploring an arcane legacy process, but it's easy to qualify it as valuable. There is knowledge (literally) dying in a removed and mentorship-heavy industry. Without the next generation taking up the sword, we'll end up dependent on systems no one knows how to operate.

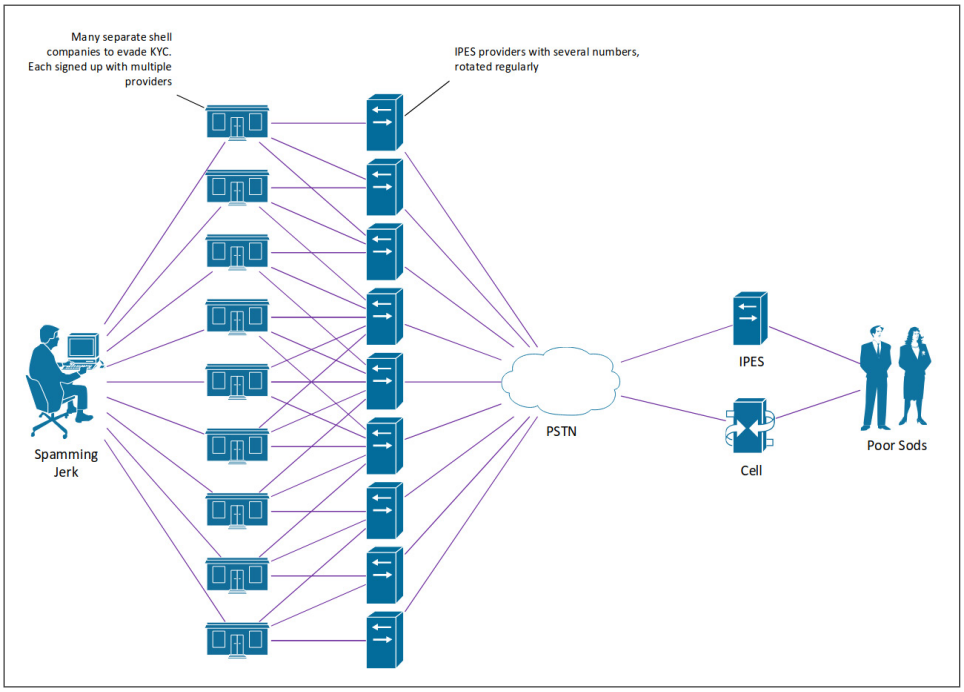


Figure 15. *A diagram showing how robocalling spammers use the organisationally-disaggregated PSTN network to continue to spam and scam.*

Safe Harbor or Hostile Waters: Unveiling the Hidden Perils of the TorchScript Engine in PyTorch

***Authors**: Ji'an Zhou and Lishuo Song*

Slides

The loading of untrusted "pickles" has been known to be a security risk for years. PyTorch, the popular Python-based ML framework, uses the pickle format to serialise its models. The community was alerted to the security risks of loading untrusted models, so a restricted subset of pickle was used (when passed `weights_only=True`) that can only load the numerical weights from a model.

This research shows that, even with the restricted loading option, there is a PyTorch-specific language environment, called TorchScript, that works to marshal a model into the correct format in a platform-agnostic manner. A specially-crafted model file can insert TorchScript instructions to gain an arbitrary file read and write primitive. With the ability to write to, for example, the `.zshrc` file, command execution is then possible. After this vulnerability was reported and fixed, the researchers explored the sprawling ecosystem of ML frameworks that rely on PyTorch under the hood, many with fixed import versions, preventing fixes from propagating.

TAKEAWAYS:

- **It is hard to miss the parallels** vis-à-vis security issues between the ML and software development communities. Only recently have we begun grappling with the consequences of cheap dependencies and opaque software supply chains. With centralised repositories like GitHub, npm, and PyPI, it is a single line to massively expand the amount of code that is part of an application.
- **This research,** and the preceding work to raise awareness of the dangers of pickles, shows that HuggingFace provides the same ease to ingest and operate on untrusted assets. While this vulnerability has been fixed, there will be a long tail of frameworks that haven't updated or moved to a safer format for exchanging models.
- **Only recently have scanners** for CI/CD jobs and code packages started to operate and find malice at scale (such as the recent tj-actions/changed-files incident). An analogous infrastructure is needed to handle the rapid publication and dissemination of ML models to ensure that they are not malicious, and to look for backdoors added to the models themselves.

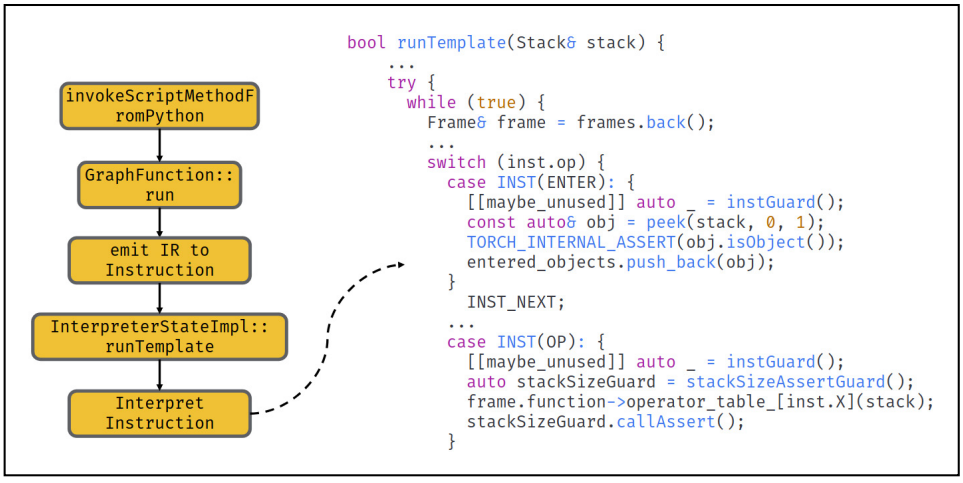


Figure 16. *Flow of how TorchScript IR is interpreted from a saved model.*

Ghosts in the Machine Check – Conjuring Hardware Failures for Cross-ring Privilege Escalation

Author:
Christopher Domas

Slides

Code

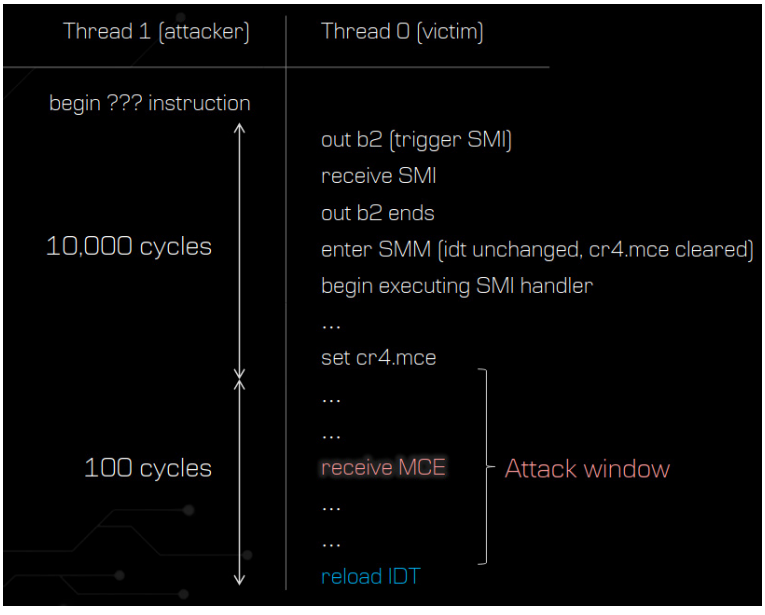
Video

This research explores how conflicting security assumptions about interrupt handling on modern systems can result in security vulnerabilities. System interrupts allow for connected devices or other system components to receive the immediate attention from the operating system when an event occurs. Depending on the type of interrupt, the OS handles it via the interrupt descriptor table (IDT). Since interrupts can occur at any point in time, there are various ways to pause or defer new interrupts from occurring to prevent system state corruption.

The researcher explored a specific interrupt that cannot be masked or otherwise deferred, the machine check exception (MCE). MCEs occur when there is a catastrophic hardware failure, temperature overrun, or other type of failure in which recovery is infeasible – the OS simply tries to limit permanent corruption and shutdown.

By fuzzing the northbridge of a modern system, a specific combination of configuration bits were discovered that would trigger a MCE when trying to access a PCIe device that did not exist. This allowed for one core of a system (the attacker) to trigger an MCE delivered to another core (the victim). However, since kernel-level permissions were needed to configure and trigger the MCE, it was not impactful without additional effort. Like past kernel-level system vulnerabilities, the researcher set their sights on system management mode (SMM, sometimes referred to as ring -2), which is a more highly-privileged mode that can access all system memory. When transition to SMM occurs, all CPU threads must enter SMM at (almost) the same time as the system management mode memory (SMRAM) is briefly unlocked. By reversing-engineering the SMM code, the researcher identified a brief period where the untrusted kernel IDT is executed while SMRAM is unlocked. Finally, the researcher discovered that an unaligned, 64-bit read of memory-mapped PCI space would take enough clock cycles that it would complete after the vulnerable window was reached, causing an MCE to be handled in SMM by the untrusted kernel handler. This gave a kernel-level attacker full access to SMM and all system memory. While the SMM handling code was changed to update the IDT with a trusted one, closing the window of vulnerability, patching SMM is a difficult and slow process.

Figure 17.
A diagram showing the attack goal of attacking SMM from another thread in a limited window where SMM accepts machine check exceptions and is using the untrusted IDT.



TAKEAWAYS:

- **The real-world security impacts** of this exploit are slight, but show a glimpse into the incredible complexity underlying modern computer systems. As noted in the conclusion, there is much left to learn about the composition of complex systems. There are so many component subsystems joined together, each with their own invariants, requirements and assumptions, that when closely examined, odd behaviours emerge.
- **Independent research** into the deepest parts of our systems is important to keep vendors honest, and to remind us all of how incredible it is that digital systems work at all.

Machine Against the RAG: Jamming Retrieval-Augmented Generation with Blocker Documents

***Authors**: Avital Shafran, Roei Schuster, and Vitaly Shmatikov*

Paper

Code

This research explored ways to alter the results of retrieval-augmented generation (RAG) LLM search systems. The goal was to "jam" the RAG system through the introduction of one attacker-controlled document (e.g., a product review, internal wiki page, etc.). This blocker document would embed close to the targeted query, and contain data that would trigger the LLM to not respond to the query.

The research explored three methods to alter the results: prompt injection in the document, using another LLM as an oracle to create the document, and a black-box optimisation approach to estimate the target LLM's embedding and similarity scoring to generate documents that overrule the others selected. In a multi-LLM evaluation, the indirect prompt injection and optimisation approaches performed best, blocking results for the targeted query up to 90% of the time, depending on LLM and composition of the other documents in the database.

When applied to the larger commercial models that are already under scrutiny for jailbreaking and injection, the attack was less successful, only jamming 30% of the queries with GPT-4o-mini and Gemini-1.5-flash. For the even more powerful GPT-4o, the attack only worked 10% of the time.

TAKEAWAYS:

- **While jamming a RAG system** is stealthy, the ability for a document to alter the output beyond simple jamming means that LLM-generated summaries (e.g., for product reviews) are at risk of manipulation. RAG is one of the areas in which LLMs have been used to provide valuable answers based on localised datasets – it's concerning that a single document addition can erase some of this value.
- **There is a strong financial incentive** to manipulate AI summaries. While the majority of the incorrect AI summaries (e.g. Google's AI headliner) have been corrupted for amusement, it will not be long until this is weaponised to alter ratings, search results, and to push disinformation and malware.

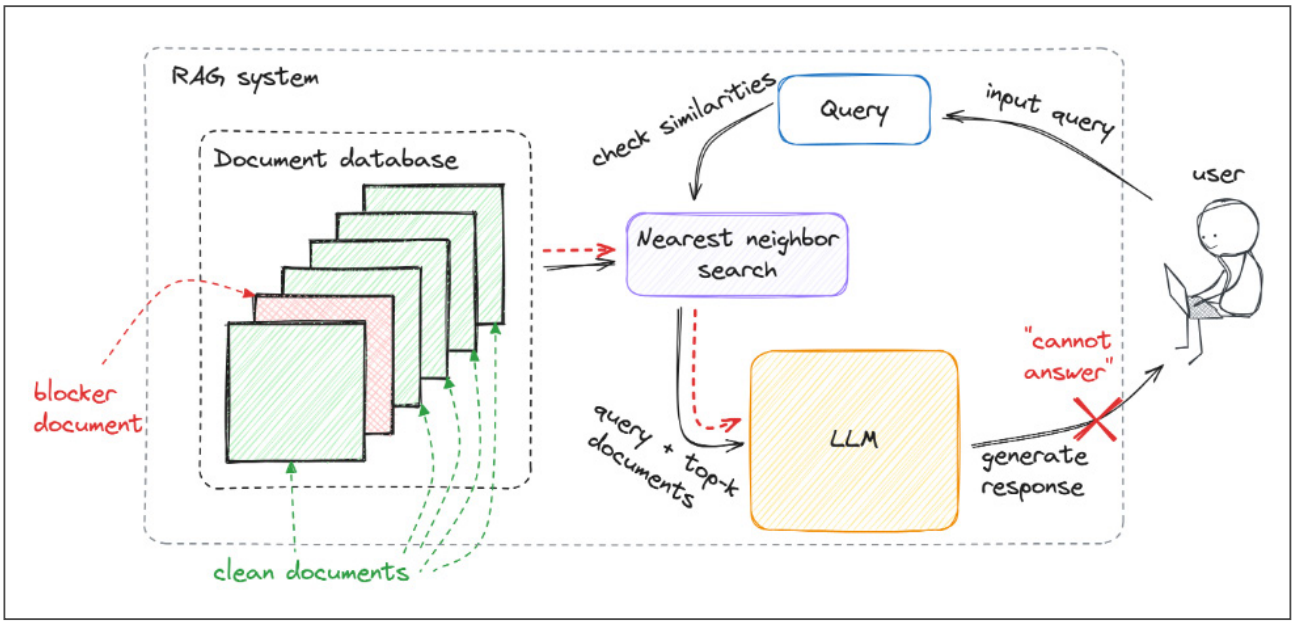


***Figure 18.** A high-level diagram showing how the blocker document inserted by an attacker convinces the LLM that it cannot answer the targeted query.*

Inverting the Xorshift128+ random number generator

***Author**: Scott Contini*

Blog

Code

This blog post examines the pseudo-random number generator (PRNG) algorithm underpinning JavaScript's `Math.random()` and finds ways to predict its output with only two or three random values. PRNGs are not supposed to be used for security-critical entropy generation, but often times are when the security relevance is less clear (i.e., outside of pure cryptographic use cases). The author came across an attack on this algorithm that, with five consecutive samples, could break the PRNG and allow for the prediction of future values.

Through an analysis of the algorithm, the blog post can improve on that result significantly – reducing the number of samples needed to two (or three for `Math.random()`'s use of the PRNG). From an observation that half of the input state data is reused, the brute forcing space is decreased from 2^{128} to 2^{64}. By examining the specific bit operations, the realisation is that only the 26 least significant bits are needed to generate the more significant bits and check the prediction. This results in a very breakable 2^{26} search space. Finally, since `Math.random()` tosses out some of the output bits, there is a higher chance of a false positive, so additional checks are needed, bringing the overall search space (with two or three consecutive values) for JavaScript to 2^{50}.

TAKEAWAY:

While this blog post focused primarily on the theoretical attacks against the PRNG itself, the implications of predicting `Math.random()` with only 2-3 previous outputs is serious. JavaScript is everywhere – look for this bug to rear its head for years to come in all sorts of places.

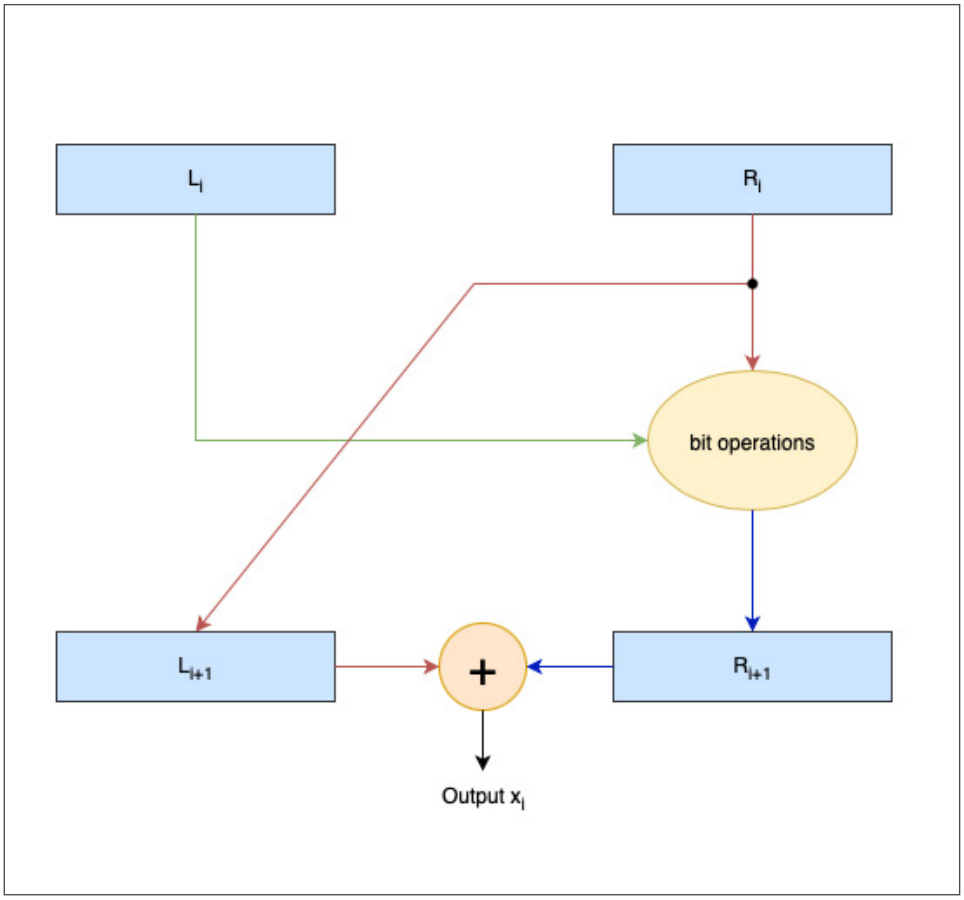


***Figure 19.** A graphical description of how each round of Xorshift128+ works.*

The Breede River, Western Cape, South Africa.
Image by Lisa van Staden (Thinkst)

Conclusions

That's all for this quarter.

WE HIGHLIGHTED FOUR THEMES THIS TIME:

1. Microsoft-induced issues.
2. Logs not always helping.
3. LLMs doing their own thing.
4. Vibration attacks.

Whew! It's been quite a quarter. Usually as we head into the holiday period there's a slight decrease in content volume, but we'll be back next time to highlight the community's work.


THINKST
LABS