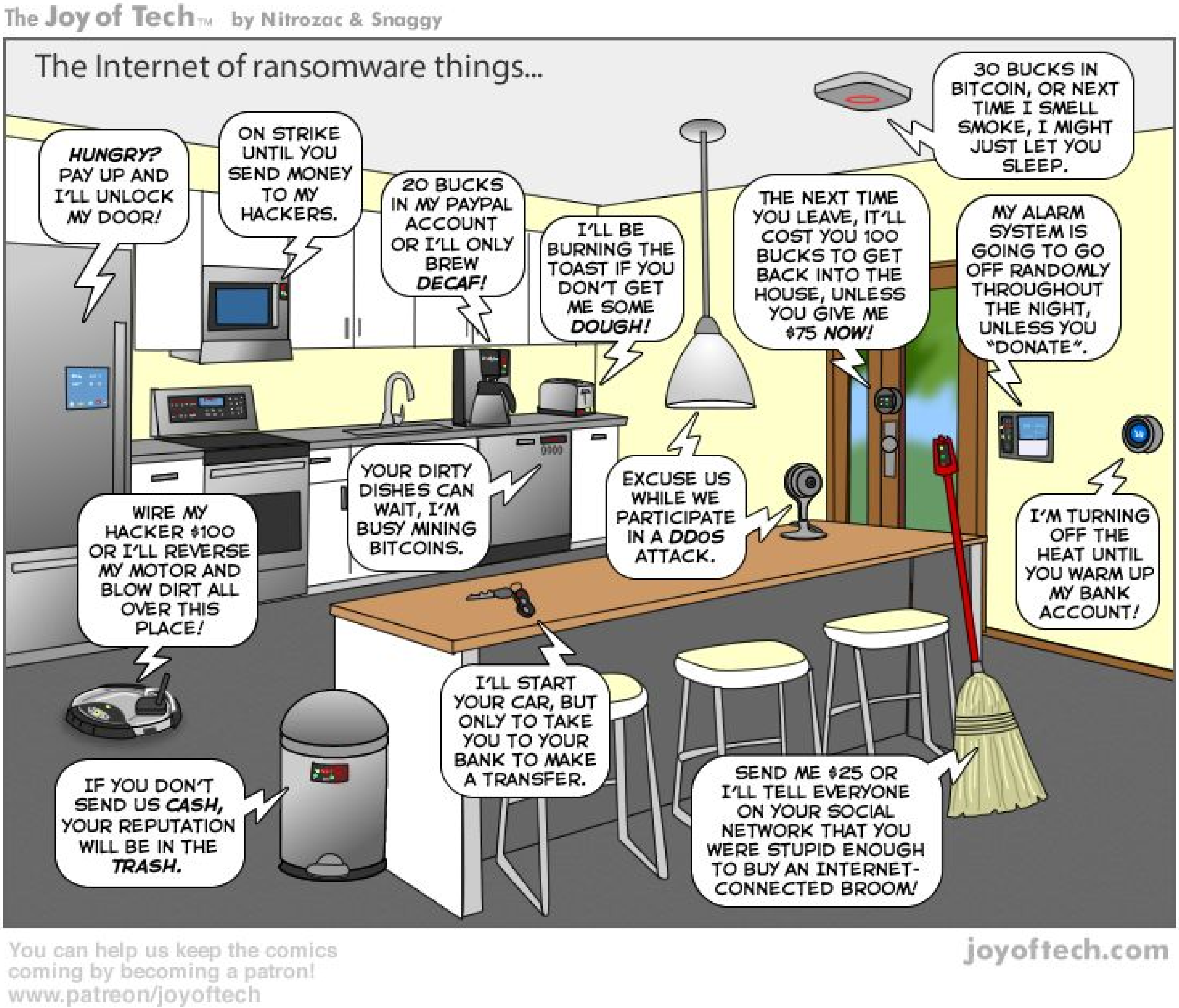
Security Now! #583 - 10-25-16

DRAMMER

This week on Security Now!

Is the Internet still working after last week's powerful attack?, Linux's worrisome "Dirty COW" bug rediscovered in the kernel after nine years, A look at the worrisome average lifetime of Linux bugs, A small bit of errata and miscellany, And an in-depth analysis of DRAMMER, the new, unpatachable, Android mobile device Rowhammer 30-second exploit.


The Joy of Tech™ by Nitrozac & Snaggy
The Internet of ransomware things...
HUNGRY? PAY UP AND I'LL UNLOCK MY DOOR!
ON STRIKE UNTIL YOU SEND MONEY TO MY HACKERS.
20 BUCKS IN MY PAYPAL ACCOUNT OR I'LL ONLY BREW DECAF!
I'LL BE BURNING THE TOAST IF YOU DON'T GET ME SOME DOUGH!
THE NEXT TIME YOU LEAVE, IT'LL COST YOU 100 BUCKS TO GET BACK INTO THE HOUSE, UNLESS YOU GIVE ME $75 NOW!
30 BUCKS IN BITCOIN, OR NEXT TIME I SMELL SMOKE, I MIGHT JUST LET YOU SLEEP.
MY ALARM SYSTEM IS GOING TO GO OFF RANDOMLY THROUGHOUT THE NIGHT, UNLESS YOU "DONATE".
YOUR DIRTY DISHES CAN WAIT, I'M BUSY MINING BITCOINS.
EXCUSE US WHILE WE PARTICIPATE IN A DDoS ATTACK.
I'M TURNING OFF THE HEAT UNTIL YOU WARM UP MY BANK ACCOUNT!
WIRE MY HACKER $100 OR I'LL REVERSE MY MOTOR AND BLOW DIRT ALL OVER THIS PLACE!
I'LL START YOUR CAR, BUT ONLY TO TAKE YOU TO YOUR BANK TO MAKE A TRANSFER.
IF YOU DON'T SEND US CASH, YOUR REPUTATION WILL BE IN THE TRASH.
SEND ME $25 OR I'LL TELL EVERYONE ON YOUR SOCIAL NETWORK THAT YOU WERE STUPID ENOUGH TO BUY AN INTERNET-CONNECTED BROOM!
You can help us keep the comics coming by becoming a patron! www.patreon/joyoftech
joyoftech.com

Security News

Attack of the Lightbulbs (All your lightbulbs are belong to us.)

- My own experience:
    - I first became aware of the problem when Sue, who was away travelling with her laptop shot me an iMessage saying that Eudora was returning a "grc.com address not resolved" error.
    - Then GRC's eCommerce system began failing to contact our upstream merchant gateway.
    - Once I learned that the trouble was with DNS, I looked up the merchant gateway IP from home (COX was resolving it) and I quickly added a HOST entry into GRC's server which, as we know, takes immediate precedence over DNS lookups... and purchases were immediately restored.

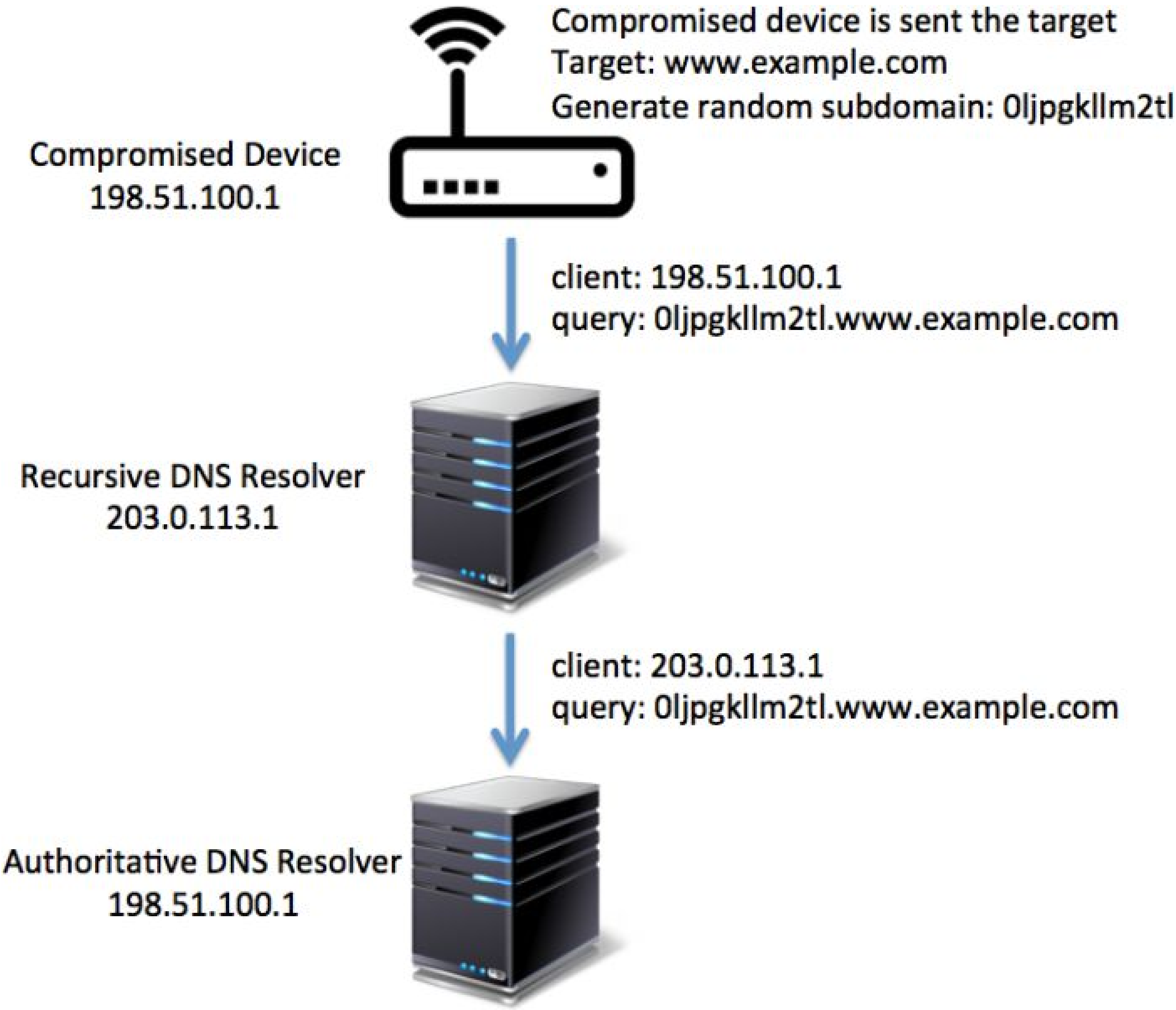
- Dyn Statement on 10/21/2016 DDoS Attack
    - http://dyn.com/blog/dyn-statement-on-10212016-ddos-attack/
    - What We Know:
      At this point we know this was a sophisticated, highly distributed attack involving 10s of millions of IP addresses. We are conducting a thorough root cause and forensic analysis, and will report what we know in a responsible fashion. The nature and source of the attack is under investigation, but it was a sophisticated attack across multiple attack vectors and internet locations. We can confirm, with the help of analysis from Flashpoint and Akamai, that one source of the traffic for the attacks were devices infected by the Mirai botnet. We observed 10s of millions of discrete IP addresses associated with the Mirai botnet that were part of the attack.

- BCP 38 and SAVE: Source Address Validation Everywhere.
    - http://www.bcp38.info/
    - BCP38 is RFC2827: Network Ingress Filtering: Defeating Denial of Service Attacks which employ IP Source Address Spoofing.

- The Mirai attacked DynDNS with a weaponized version of the technique I used six years ago for GRC's DNS Benchmark and GRC's DNS Spoofability testing:

    - In both cases I use this to GENTLY bust the cache in order to force fresh DNS lookup resolutions from servers are different stages in the DNS caching hierarchy. I was always, of course, careful to meter my queries over time so as to not overload the target servers. And since I also wanted to determine resolution reliability, I explicitly didn't want to saturate any connections and cause false positive packet losses.

- Mirai has a different purpose in mind.
    - Mirai prepends a pseudorandom 12 character subdomain onto the target domain.
    - Some attacking devices were issuing 2500 queries per second.


Compromised device is sent the target
Target: www.example.com
Generate random subdomain: 0ljpgkllm2tl
Compromised Device
198.51.100.1
client: 198.51.100.1
query: 0ljpgkllm2tl.www.example.com
Recursive DNS Resolver
203.0.113.1
client: 203.0.113.1
query: 0ljpgkllm2tl.www.example.com
Authoritative DNS Resolver
198.51.100.1

- DNS Caching and cache draining
  - Standard DNS cache time
  - Why higher, why lower?

- As Bruce Schneier recently explained about IoT-device security:
  "The market can't fix this because neither the buyer nor the seller cares."

- http://iotscanner.bullguard.com/

"Dirty COW" - Linux Privilege escalation bug

- https://dirtycow.ninja/
- A "High Severity" nine-year-old bug in the Linux kernel has recently come to light.
- Any system running Linux on a web facing server is vulnerable.
- Phil Oester, a Linux developer, routinely logs all HTTP traffic to his webservers and analyzes them for forensic purposes... just to keep an eye on what's going on... and he discovered this flaw being deployed against his servers in the wild!

- This flaw:
  - Allows an attacker to gain write access to read-only memory.
  - Is not difficult to develop exploits for that work reliably.
  - Is located in a section of the Linux kernel that's a part of virtually every distribution of the open-source OS released for the last nine years.

- Named "Dirty COW" with "COW" standing for Copy on Write.
- Code introduced into the kernel 9 years ago contains a "race condition"

- Linus Torvalds <torvalds@linux-foundation.org>
  - https://lkml.org/lkml/2016/10/19/860
  - Thu, 20 Oct 2016 00:49:40 +0200
    This is an ancient bug that was actually attempted to be fixed once (badly) by me eleven years ago in commit 4ceb5db9757a ("Fix get_user_pages() race for write access") but that was then undone due to problems on s390 by commit f33ea7f404e5 ("fix get_user_pages bug").

- Dirty COW can be used in a hosted web environment where shell access is provided to allow one customer to attack others.

- Whereas an SQL injection weakness would normally allow attacker's code to run unprivileged, combining the two bugs would allow an attacker to run their code with root privilege.

- The vulnerability is easiest exploited with local access to a system, such as shell accounts, where an attacker can obtain root in fewer than 5 seconds.

- Linux kernel maintainers have released the patch and users are advised to install it as soon as possible.

Google's Kees (pronounced 'case') Cook analyzes the lifetime of Linux bugs...

- He describes himself:
  I work for Google on ChromeOS security. Previously, I worked for 5 years at Canonical as an Ubuntu Security Engineer. My work is to stay alert, curious, and creative while keeping one step ahead of the bad guys. When I'm not working, I've been known to play with MythTV and generally poke around at video formats.

- https://outflux.net/blog/archives/2016/10/18/security-bug-lifetime/

- <quote> In several of my recent presentations, I've discussed the lifetime of security flaws in the Linux kernel. Jon Corbet did an analysis in 2010, and found that security bugs appeared to have roughly a 5 year lifetime. As in, the flaw gets introduced in a Linux release, and then goes unnoticed by upstream developers until another release 5 years later, on average. I updated this research for 2011 through 2016, and used the Ubuntu Security Team's CVE Tracker to assist in the process. The Ubuntu kernel team already does the hard work of trying to identify when flaws were introduced in the kernel, so I didn't have to re-do this for the 557 kernel CVEs since 2011.

- The numerical summary is:
  - Critical: 2 @ 3.3 years
  - High: 34 @ 6.4 years
  - Medium: 334 @ 5.2 years
  - Low: 186 @ 5.0 years

- This comes out to roughly 5 years lifetime again, so not much has changed from Jon's 2010 analysis.

- <quote> While we're getting better at fixing bugs, we're also adding more bugs. And for many devices that have been built on a given kernel version, there haven't been frequent (or some times any) security updates, so the bug lifetime for those devices is even longer. To really create a safe kernel, we need to get proactive about self-protection technologies. The systems using a Linux kernel are right now running with security flaws. Those flaws are just not known to the developers yet, but they're likely known to attackers, as there have been prior boasts/gray-market advertisements for at least CVE-2010-3081 and CVE-2013-2888.

Errata

Tech luminaries laud Dennis Ritchie .... 5 years after death

- https://www.cnet.com/news/tech-luminaries-laud-dennis-ritchie-5-years-after-death-second-death-syndrome/
- SubTitle: 'Well-known tech figures appear to have forgotten the father of the C programming language died years ago, falling victim to social media's "second death syndrome."'

- Some of tech's biggest names are paying tribute this evening to computing pioneer Dennis Ritchie.

  Ritchie was an internationally renowned computer scientist who created the C programming language. He also made significant contributions to the development of the Unix operating system, for which he received the Turing Award in 1983.

  The problem, especially if you look at it from Ritchie's perspective, is that he's been dead for five years -- exactly five years. That time gap seems to have escaped some of the biggest names in tech, including Google CEO Sundar Pichai, who late Wednesday tweeted out Wired's five-year-old obituary on Ritchie, thanking him for his "immense contributions."

Om Malik, a partner at True Ventures and the founder of tech site GigaOm, retweeted Pichai's tribute before soon recognizing his mistake and tweeting an apology for "adding to the confusion and noise."

Miscellany

AT&T / Time Warner merger?

- AT&T has offered $85.4 billion for Time Warner, which would give it control of, among other media goodies, HBO, CNN, the Warner Bros. studio and DC Comics.

This Week in Energy Storage

- Title: "Scientists Accidentally Discover Efficient Process to Turn CO2 Into Ethanol"
  - http://www.popularmechanics.com/science/green-tech/a23417/convert-co2-into-ethanol/

- The process is cheap, efficient, and scalable, meaning it could soon be used to remove large amounts of CO2 from the atmosphere.

- Oak Ridge National Laboratory
  - https://www.ornl.gov/news/nano-spike-catalysts-convert-carbon-dioxide-directly-ethanol

- Title: "Nano-spike catalysts convert carbon dioxide directly into ethanol"
  - http://onlinelibrary.wiley.com/doi/10.1002/slct.201601169/full
  - http://onlinelibrary.wiley.com/doi/10.1002/slct.201601169/pdf

- <QUOTE>
  OAK RIDGE, Tenn., Oct. 12, 2016—In a new twist to waste-to-fuel technology, scientists at the Department of Energy’s Oak Ridge National Laboratory have developed an electrochemical process that uses tiny spikes of carbon and copper to turn carbon dioxide, a greenhouse gas, into ethanol. Their finding, which involves nanofabrication and catalysis science, was serendipitous.
  ORNL’s Adam Rondinone, lead author of the team’s study published in Chemistry Select, said: “We discovered, somewhat by accident, that this material worked. We were trying to study the first step of a proposed reaction when we realized that the catalyst was doing the entire reaction on its own!”

  The team used a catalyst made of carbon, copper and nitrogen and applied voltage to trigger a complicated chemical reaction that essentially reverses the combustion process. With the help of the nanotechnology-based catalyst which contains multiple reaction sites, the solution of carbon dioxide dissolved in water turned into ethanol with a yield of 63 percent. Typically, this type of electrochemical reaction results in a mix of several different products in small amounts.

Adam continued: “We’re taking carbon dioxide, a waste product of combustion, and we’re pushing that combustion reaction backwards with very high selectivity to a useful fuel. Ethanol was a surprise -- it’s extremely difficult to go straight from carbon dioxide to ethanol with a single catalyst. But this does that.”

- The system operates by dissolving CO2 in water, at room temperature, and, with a bit of electricity, produces ethanol.

Peter Hamilton's "A Night Without Stars"

- At 24%... in "Book Three" ... WHAM!! <g>

SpinRite

Colin Wills in Dorking, England
Subject: Spinrite saves a guitar lesson :-)
:
I worked from home today using my bitlockered Windows 7 work laptop so that I could take my daughter to her guitar lesson. But last night I thought I was going to have to go in to the office and that we would have to skip the lesson because my laptop wouldn't boot. It had recently hung at some point during the boot but had recovered at a second attempt. But last night it repeatedly got stuck. Being a Security Now listener almost from the start I immediately thought of Spinrite. (Actually started around Snowden but also recapped & now just past the sugar hill at 353 and reading Taubes.) The work laptop is locked down such that I cannot boot my Spinrite CD so I transferred the HD to my personal laptop and Spinrote it quickly at level 2. Of course the fact it was bitlockered didn't matter and Spinrite cut through the disk like a knife through full fat butter and completed after about an hour. After that the work laptop booted, I worked from home and we made it to the guitar lesson.

Thanks for the great product. I am really really looking forward to v6.1!
All the best to you and Leo,
Colin.

DRAMMER (DRAM Hammer)

A team of security researchers at VU University in Amsterdam, led Professor Herbert Bos, in concert with some researchers at UC Santa Barbara decided to explore the possibility of inducing their Flip Feng Shui DRAM bit flipping on ARM-based Android mobile devices.
The result is major panic at Google Central... because they arrived at an exploit that relies upon NO FLAWS in the OS while delivering full root privilege to an initially unprivileged app in roughly 30-seconds... and there's no simple fix for this!

- Drammer explainer:
  - https://www.vusec.net/projects/drammer/
- Drammer test app:
  - https://vvdveen.com/drammer/drammer.apk

- Review: Flip Feng Shui

- DRAMMER:
  - Deterministic Row Hammer
  - DRAM Hammer

- ABSTRACT excerpt:
  We show that deterministic Rowhammer attacks are feasible on commodity mobile platforms and that they cannot be mitigated by current defenses. Rather than assuming special memory management features, our attack, Drammer, solely relies on the predictable memory reuse patterns of standard physical memory allocators. We implement Drammer on Android/ARM, demonstrating the practicability of our attack, but also discuss a generalization of our approach to other Linux-based platforms.

  To support our claims, we present the first Rowhammer-based Android root exploit relying on no software vulnerability, and requiring no user permissions. In addition, we present an analysis of several popular smartphones and find that many of them are susceptible to our Drammer attack. We conclude by discussing potential mitigation strategies and urging our community to address the concrete threat of faulty DRAM chips in widespread commodity platforms.

- Mobile platforms mostly use ARM processors. However, all previous Rowhammer techniques target x86 and do not readily translate to ARM.

- Researchers have questioned whether memory chips on mobile devices are susceptible to Rowhammer at all, and whether the ARM memory controller permits sufficiently rapid access to trigger bit flips.

- Drammer is an instance of the more general Flip Feng Shui (FFS).

- For any Flip Feng Shui to be successful, three requirements must be met:
    - Attackers need to be able to "hammer suciently hard", hitting the memory chips with high frequency because no bits will flip if the memory controller is too slow
    - Physical memory needs to be manipulatable, and manipulated, so that exploitable data is located in the vulnerable physical page.
    - Physical memory mapping must be determinable to allow double-sided Rowhammering, which yields many more bit flips in much less time when searching for memory pages containing vulnerable bits.

- NONE of these requirements were initially met with existing Rowhammer techniques.

- Existing hammer code usually consists of a small loop which Reads a value from memory, and flushes this value from the CPU cache. Without cache flushing, the vulnerable DRAM chip would only be accessed once during the first iteration. Every subsequent access would be served from the cache.

- With an ARMv7-A CPU, cache flush instructions are unavailable from user space (unlike Intel's clflush instruction).

- The research, therefore, first implemented a hammer function as Loadable Kernel Module. This allowed them to use the equivalent ARM cache flushing instruction... and they found their first ARM-based bit flip. After that, they found many more.

- But that was only a proof of concept, since they had cheated by first giving themselves kernel privilege. So how to get the equivalent of a cache flush to allow high-speed row hammering without access to the privileged cache flush instruction?

- DMA to the rescue... sorta.
    - By design, DMA memory pages have two important properties: 1) they are marked as uncached; and 2) they are physically contiguous.
    - Since they are uncached by design, that means that any read operation from DMA memory -- or any write operation to it -- bypasses the CPU cache and propagates to the DRAM chip directly.
    - The physical contiguity of allocated pages guarantees that pages obtained by allocations up to a certain size (4MB on recent Linux kernels) are guaranteed to be adjacent to each other in physical memory, which is a useful property for their needs.

- Exploitation
    - A large physically contiguous block of memory is allocated.

    - It is then deallocated and immediately many tiny blocks of memory are allocated.

    - This deterministically fills the large region with a combination of memory management page tables and the tiny chunks of memory those page tables describe and manage.

- The page management tables describe memory OWNED by the attacking process.

- Row Hammer is then employed to force a bit flip in one of those page tables to map one of the normally off-limits system-managed page tables into the visible and accessible memory space of the attacking process.

- So this has allowed the process to obtain read/write access to its own memory management structures, which are, by design, absolutely off-limits to OS client processes.

- By manipulating its own page table, the process can map ANY physical memory into its process memory space.

- So they search for their own process' Linux process credentials structure which the kernel uses to keep track of UID, GID, etc. Once found... they simply alter their own credentials to give themselves root privilege.

- The exploit is fully reliable and may only fail if the flip discovered in the templating phase is not reproducible.