Theoretical Attacks to FIDO2/Yubikey

Introduction

When signing in with your ID (Azure AD) identity through a browser, Microsoft provides a wide range of excellent options for you to choose from as your primary authentication method.

- Username and Password
- Phone number and SMS
- Username and Passwordless phone sign-in
- Certificate-based authentication
- FIDO2 security keys

By leveraging conditional access, you have the ability to enhance the security of your accounts. This includes implementing additional measures such as requiring a second factor of authentication, ensuring device compliance, imposing location-based restrictions, and configuring numerous other options for added protection.

Phishing

It is very important to know that only two methods mentioned protect your users against phishing attacks.

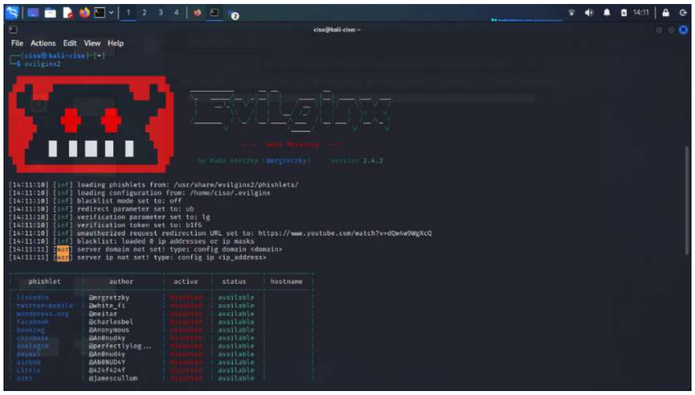
- **Windows Hello for Business**
- **FIDO2 security keys**

To understand why this is the case, it is most effectively examined by utilizing one of the readily accessible phishing toolkits like Evilginx.

As the saying goes, "seeing is believing," and I find that hands-on exploration is the most effective way for me to learn and comprehend the subject matter.

Sign-in using username, password and 2FA

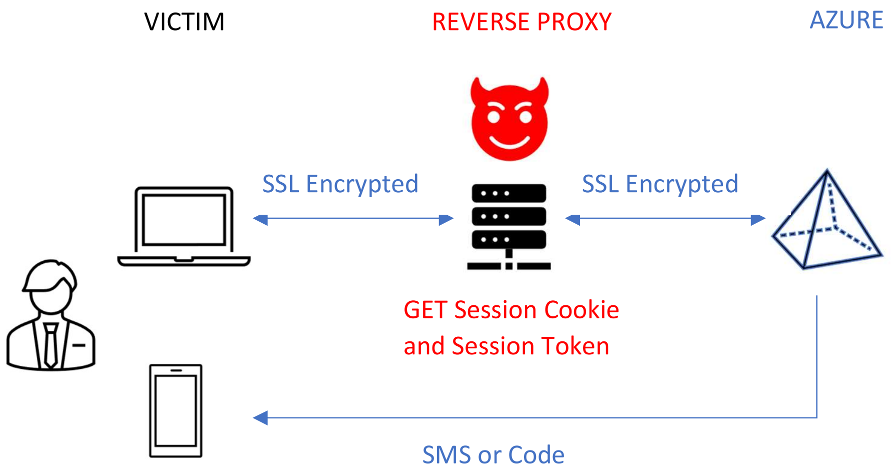
Upon clicking the link, the victim is directed to a login form for ID (Azure AD). They enter their username and password as requested and subsequently receive a pop-up notification on the Microsoft Authenticator app. Since the victim anticipates this step, they approve the sign-in request, leading to successful login. However, instead of accessing the intended document, the victim is redirected to www.office.com.



```
┌──(ciso㉿kali-ciso)-[~]
└─$ evilginx2

                    - --  Next Phishing  -- -

      by Kuba Gretzky (@mrgretzky)      version 2.4.2

[14:11:10] [inf] loading phishlets from: /usr/share/evilginx2/phishlets/
[14:11:10] [inf] loading configuration from: /home/ciso/.evilginx
[14:11:10] [inf] blacklist mode set to: off
[14:11:10] [inf] redirect parameter set to: ub
[14:11:10] [inf] verification parameter set to: lg
[14:11:10] [inf] verification token set to: b1f6
[14:11:10] [inf] unauthorized request redirection URL set to: https://www.youtube.com/watch?v=dQw4w9WgXcQ
[14:11:10] [inf] blacklist: loaded 0 ip addresses or ip masks
[14:11:11] [war] server domain not set! type: config domain <domain>
[14:11:11] [war] server ip not set! type: config ip <ip_address>

    phishlet     |     author       |  active   |  status   | hostname
-----------------+------------------+-----------+-----------+----------
 linkedin        | @mrgretzky       | disabled  | available |
 twitter-mobile  | @white_fi        | disabled  | available |
 wordpress.org   | @meitar          | disabled  | available |
 facebook        | @charlesbel      | disabled  | available |
 booking         | @Anonymous       | disabled  | available |
 coinbase        | @An0nud4y        | disabled  | available |
 onelogin        | @perfectlylog ...| disabled  | available |
 paypal          | @An0nud4y        | disabled  | available |
 airbnb          | @AN0NUD4Y        | disabled  | available |
 citrix          | @424f424f        | disabled  | available |
 o365            | @jamescullum     | disabled  | available |
```

Photo: Evilginx / Open Source Reverse Proxy for Phishing

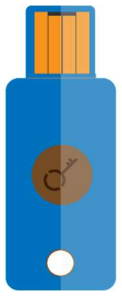
Typically, the attacker already possesses the username (i.e email address) and only needs to gather the session cookie and session token. By utilizing the "sessions" command in evilginx, they can examine and even extract the session cookie, storing it for future purposes. A similar scenario can arise with "Passwordless" Authentication, where the cookie and token can be acquired and exploited to gain unauthorized access to the system.



Secure Authentication: Sign-in using FIDO2 security keys

The user using an external FIDO2 security key has a greater protection. For example, the user will be denied access to the phishing website, which is a mirrored replica of the real one (ex. Microsoft Login Form). Since the user is unable to log in, the session key will not be created and remain uncompromised.

FIDO2/WebAuthn employs various techniques to prevent phishing of credentials. One particularly effective method involves preventing users from signing in through a straightforward yet highly impactful technique.



The WebAuthn Client (the browser) compares the domain name with the Relying Party Identifier (RP ID) found in the public keys of the FIDO2 security key.

If the domain string matches, it serves as a valid method for signing in. On the other hand, if the domain string does not match, the user is unable to utilize the credential stored in the key. This protection mechanism is exceptionally reliable because machines excel at comparing string values, making it nearly impossible for even a "clever" phishing attack using an Internationalized Domain Name (IDN) that appears identical to the target domain to bypass this safeguard.

Why FIDO2 is phishing resistant?

Origin checking is a crucial aspect of the security mechanisms in FIDO2, providing strong protection against phishing attacks. Here's a more detailed explanation of how origin checking works and why it's so effective:

Origin checking is a process where the FIDO2 authenticator (such as a security key or a device with biometric sensors) verifies the authenticity of the website or service requesting authentication. This verification is done by comparing the origin of the request with the known, registered origin.

In web contexts, an origin typically consists of the scheme (such as https), the host (like example.com), and the port. These components uniquely identify a web service or site.

How Origin Checking Works in FIDO2:

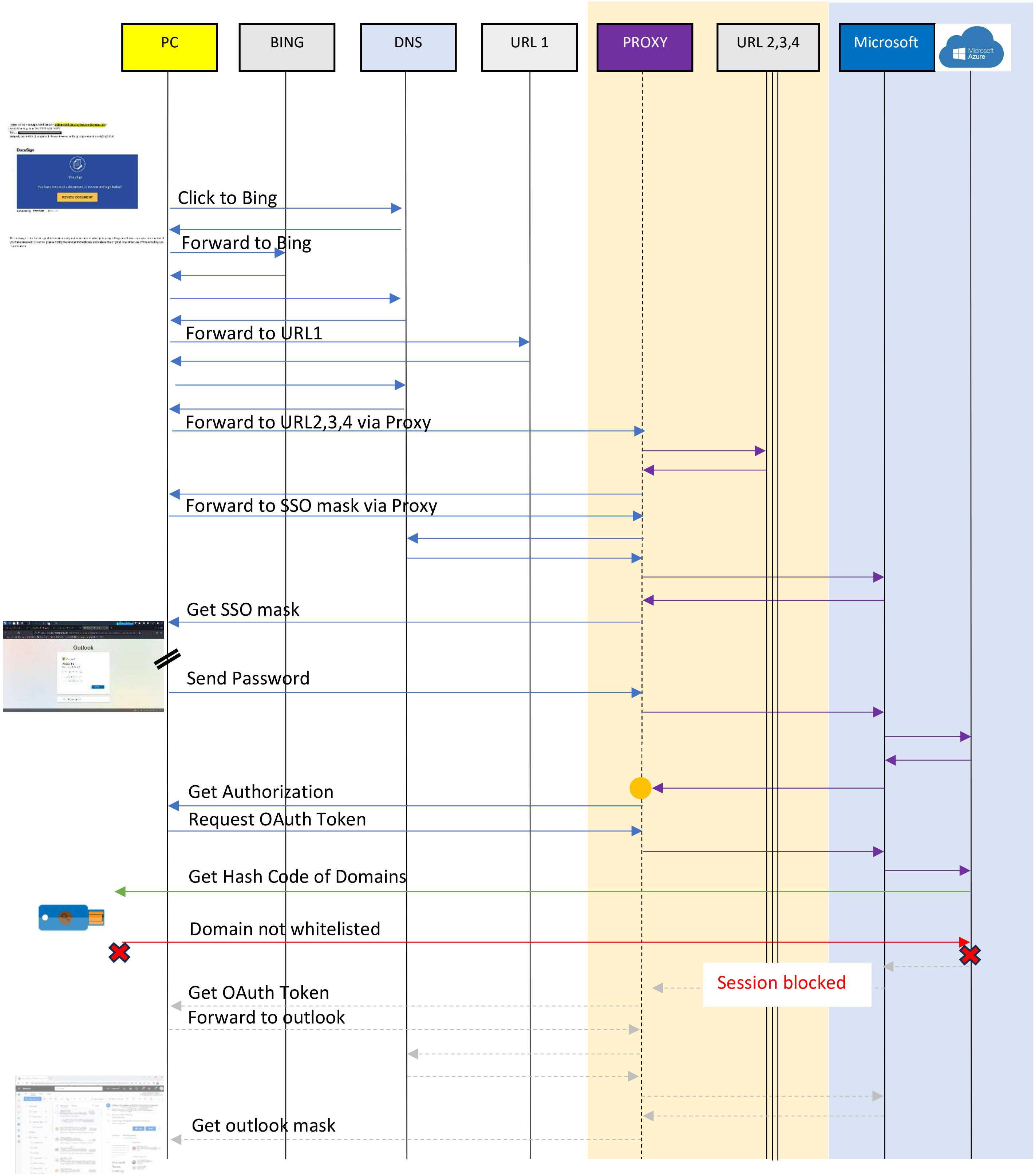
- Registration Phase: When you first register a FIDO2 device with a service (like an online account), the device stores the origin (such as https://example.com) along with the public key in its secure storage. This pairing of the origin with the key is crucial for later verification.
- Authentication Phase: When you attempt to log in to the service later, the service sends an authentication challenge to your FIDO2 device. The device then checks the origin of this request. If the origin matches what was stored during registration, the device proceeds to sign the challenge with its private key.

If the origin does not match (indicating a potential phishing attempt), the device refuses to authenticate the request.

Why Origin Checking is Effective Against Phishing

- Prevents Mimicking: Phishing sites often try to mimic the login pages of legitimate services. However, they cannot mimic the origin because they cannot replicate the exact domain and protocol of the legitimate service.
- Automatic and Transparent: This check is done automatically and transparently to the user. The user doesn’t need to verify the URL or take any additional steps.
- No Room for User Error: Since the check is done by the device and not the user, it eliminates the risk of user error, such as failing to notice a slightly misspelled URL or a wrong domain.
- Integral Part of Cryptographic Challenge: The origin is an integral part of the cryptographic challenge-response process in FIDO2. The device only signs the challenge if the origin is correct, ensuring the response is only valid for the intended service.

MFA Phishing Resistant with FIDO2 - Simplified TCP-UDP/DNS Protocol Flow – reverse Engineering (probable TCP flow)



Theoretical attacks to FIDO2

FIDO2 is considered highly secure due to its advanced features and protocols designed to enhance online authentication:

- Public Key Cryptography: FIDO2 employs public key cryptography, where the private key remains securely on the user's device, significantly reducing the risk of key theft or interception.
- Biometric Authentication: It supports biometric authentication (like fingerprints or facial recognition), which adds a layer of security by tying access to the user's unique physical characteristics.
- No Shared Secrets: Unlike password-based systems, FIDO2 doesn't rely on shared secrets that can be easily compromised. This design minimizes the risk of large-scale data breaches.
- Protection Against Phishing and Man-in-the-Middle Attacks: By using direct communication between the authenticator and the website, FIDO2 helps prevent phishing and man-in-the-middle attacks, common vulnerabilities in traditional authentication methods.

In summary, FIDO2's use of cutting-edge cryptographic methods, support for biometric verification, and resistance to common cyber threats make it a highly secure framework for online authentication.

However, the YubiKey can be configured for two-factor authentication, supporting authenticator codes such as HOTP and TOTP (based on RFC 4226):

- HOTP (HMAC Based One-Time-Password): the counter is stored on the device and the OTP is transmitted via HID USB interface.
- TOTP (Time Based One-Time-Password: relies on the current time as its counter. YubiKey devices are usually passive and lack an internal Real-Time Clock (RTC) chip or a battery to power a system clock and consequently are using the clock of the device.

TOTP, or Time-Based One-Time Password, is a variation of the one-time password (OTP) algorithm that, unlike HOTP (HMAC-based One-Time Password), generates passwords based on a time factor instead of a counter. TOTP is widely used in two-factor authentication (2FA) systems and offers a dynamic passcode that changes at fixed intervals, enhancing security.

The TOTP generates an OTP by combining a shared secret key with the current timestamp, usually rounded down to a set period (like 30 or 60 seconds). This time factor means that each OTP is valid only for a short window, after which a new OTP is generated.

And the HMAC Algorithm (Hash-Based Message Authentication Code) is used in both mechanisms (HTOP and TOTP), however the counter in the HMAC algorithm is replaced with the current time.

For TOTP functionality, an external application capable of reading OATH codes from YubiKeys is essential, given that YubiKeys do not possess an internal clock. This requirement is addressed by the Yubikey Authenticator app, which conveys timestamp information from its host system to the YubiKey and retrieves the generated OTP from the YubiKey's chip.

Reflecting on the prerequisites some attacks are possible considering the protocol on page 3.

Attack 1: Replay Attack using FIDO2 USB Token

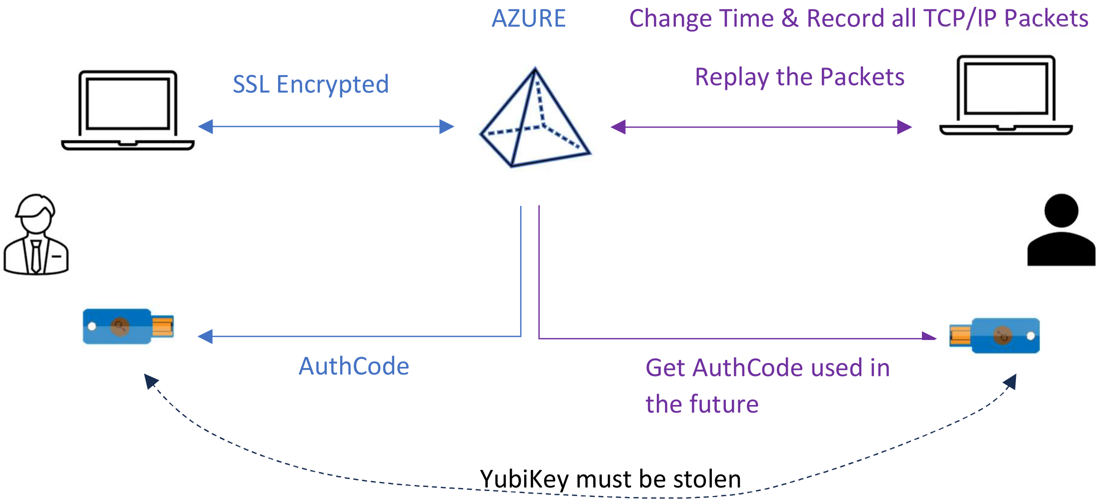
Prerequisites:

- The attacker must be aware of the victim's primary authentication details (username and password).
- The attacker requires access to the victim's U2F key for a considerable duration (e.g., over 30 minutes to record all TCP/IP Packets).

The vulnerability in question unfolds as follows:

In a targeted cyberattack, the attacker first acquires the victim's YubiKey, the physical security device. They then install the YubiKey Authenticator on a computer and advance the system time to a future date, using the YubiKey to generate numerous One-Time Passwords (OTPs) for that specific time, such as 13:00 on the 24th of December. This OTP generation process can be automated for efficiency.

After recording these future-dated OTPs, the attacker discreetly returns the YubiKey to the victim, maintaining the victim's unawareness of the breach. With the pre-generated OTPs, the attacker is poised to exploit a vulnerability in the two-factor authentication system at the predetermined future time. By logging in with these OTPs, the attacker can compromise the second factor of authentication, bypassing security and gaining unauthorized access. This sophisticated attack highlights the need for more secure authentication methods to prevent such vulnerabilities.


AZURE
Change Time & Record all TCP/IP Packets
SSL Encrypted
Replay the Packets
AuthCode
Get AuthCode used in the future
YubiKey must be stolen

Attack 2: Replay Attack using Reverse Shell in a compromised device

Prerequisites:

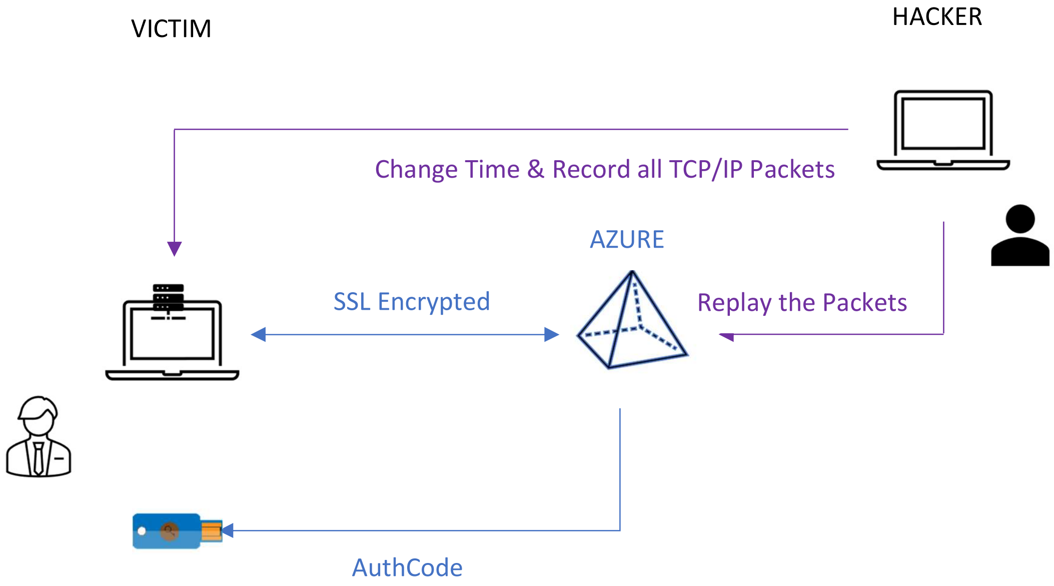
- The attacker gains control through a Reverse Shell to the device (PC) utilized for server login.

In a sophisticated cyberattack, the attacker exploits the victim's device by manipulating the device's clock and capturing encrypted network traffic. This method begins when the victim is actively using the device to login, at which point the attacker advances the device's time to a future point, such as 13:00 on the 24th of December. This time manipulation is a strategic move to exploit time-sensitive security protocols.

During this altered time period, the attacker records all incoming and outgoing encrypted traffic. After sufficient data capture, the attacker discreetly resets the device's time to its original setting to avoid detection.

The final phase involves selectively retransmitting specific captured packets, focusing on authentication processes. This allows the attacker to mimic legitimate authentication sessions, thereby bypassing security mechanisms and potentially gaining unauthorized access.

To counter such attacks, it's vital to implement robust security measures, including encryption, network monitoring, and strict time synchronization. Regular security audits and advanced intrusion detection systems are also key to mitigating these sophisticated threats, highlighting the importance of comprehensive cybersecurity in the digital age.



Attack3: Reverse Proxy in the Victim-Device

Prerequisites:

- Reverse Proxy is installed in the Victim Device

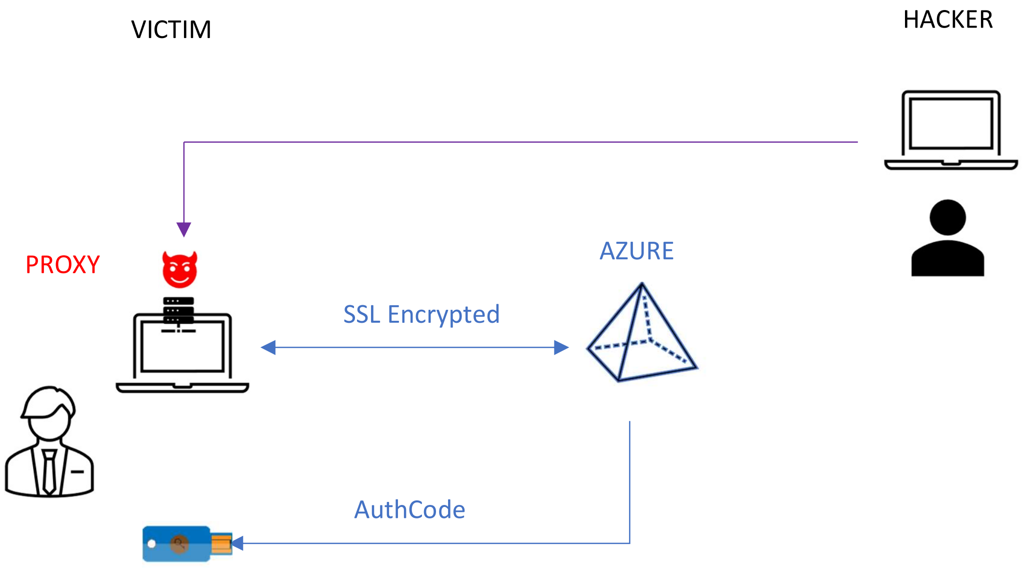
In a sophisticated cyberattack scenario, the device of an unsuspecting victim becomes compromised through the installation of a reverse proxy, which effectively reroutes the device's internet traffic directly to the attacker.

Once the reverse proxy is operational, it begins to redirect the victim's internet traffic through a path controlled by the attacker. This redirection allows the attacker to access the same resources as the victim, particularly when the victim logs into services like cloud platforms, such as Azure. Because the victim's authentication session is active, the reverse proxy relays all the necessary data, including session tokens and cookies, to the attacker.

In this compromised state, the victim's device is effectively transformed into a conduit for the attacker. It becomes an unwitting routing mechanism, allowing the attacker to invisibly piggyback on the victim's authenticated sessions. This access is especially concerning because it allows the attacker to exploit the victim's authenticated state to access sensitive resources without the need for direct authentication.

Detecting such unauthorized access is challenging, as the attacker's activity might mimic regular user behavior to evade detection. To prevent unauthorized access, it's crucial to monitor for additional, unknown sessions or connections to unrecognized IP addresses.

The implications of such a scenario underscore the importance of stringent network security measures. These measures include regular device scans for malware, the use of comprehensive security solutions, and continuous monitoring of network traffic for anomalies. Effective countermeasures might involve employing advanced intrusion detection systems (IDS), implementing strict firewall rules, and conducting regular security audits to identify and rectify vulnerabilities that could be exploited for such attacks.



Summary

Attack 1: Replay Attack using FIDO2 USB Token

Physical Access: The attacker must obtain the physical USB Key and the username & password. This step is non-trivial and typically requires direct physical access to the victim's hardware, posing a significant barrier.

TCP/IP Packet Recording: Once in possession of the USB Key, the attacker captures all TCP/IP packets transmitted between the device and the server. This process involves network sniffing or packet capturing techniques to intercept and log the data packets. The duration of this capture is critical; it must be long enough to gather sufficient data for the attack.

Objective: The primary goal here is to obtain sensitive information from the network traffic, which may include authentication data, session tokens, or other credentials. This information can be used for unauthorized access or further attacks.

Attack 2: Replay Attack using Reverse Shell in a compromised Device

Reverse Shell Implementation: This attack begins with the installation of a reverse shell on the victim's device. A reverse shell is a covert backdoor that allows an attacker to remotely execute commands on the compromised system, typically bypassing firewall restrictions.

Time Alteration: With control over the device, the attacker then changes the system's clock settings, advancing the time to a future date. This manipulation can disrupt time-sensitive security protocols, particularly those reliant on time-based one-time passwords (TOTP).

Login Attempts and Encrypted Packet Recording: The attacker then performs multiple login attempts to the targeted service, capturing all the encrypted network traffic associated with these attempts. The encrypted packets are crucial for analyzing and extracting authentication credentials or session data.

Attack3: Reverse Proxy in the Victim-Device

Reverse Proxy Installation: This strategy involves installing a reverse proxy on victim device. A reverse proxy acts as an intermediary for requests from clients seeking resources from servers. In this context, it's used maliciously to intercept and relay communications.

Access to Azure Resources: By deploying a reverse proxy, the attacker positions themselves to intercept communications between the victim and Azure services. This setup is particularly effective in cloud-based environments where resources and services are accessed remotely.

Objective and Capability: The attacker aims to gain access to resources and data within Azure whenever the victim is logged in. This approach allows for a wide range of malicious activities, including data theft, session hijacking, and further exploitation of Azure services.

In all these attacks, the combination of technical sophistication and the need for specific conditions (like physical access or successfully installing malicious software) demonstrates both the complexity and the potential severity of these security threats. Each method requires distinct skills and resources, highlighting the multifaceted nature of cybersecurity threats and the importance of robust security measures.

About Gianclaudio Moresi

A distinguished Inventor, Author, and International Speaker with over 25 years in cybersecurity, Gianclaudio Moresi has significantly influenced the field. His consulting work spans multinational corporations, governments, and military forces.

Key Publications:

- "The Three Laws of Cybersecurity" (ISBN: 978-3033087576)
- "Zero Trust Network & Zero Internet" (ISBN: 978-3033100794)

Gianclaudio's notable achievements include the development of "Zero Trust Architecture / Zero Day Protection" (2014), an algorithm for detecting "Ransomware with Intermittent Encryption" (2021), and a novel method for "MFA Bypass" detection (2023).

His contributions at CISO events have shaped global cybersecurity strategies, awareness, and risk management.