yubico

JULY 2020

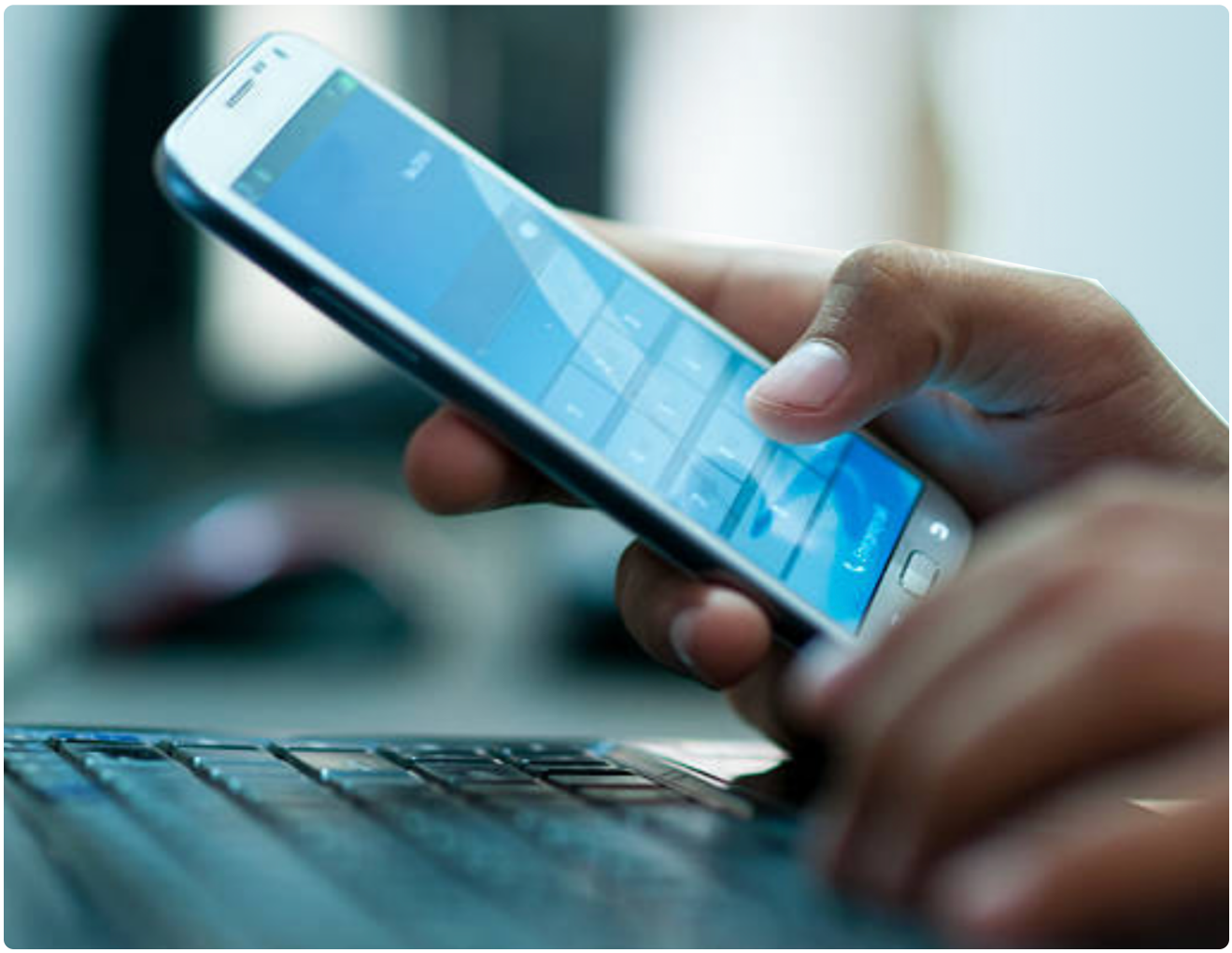
Credential Theft: Common Mitigations vs. Attacker Behaviors

Introduction

In this brief we evaluate a few common internet credential theft mitigations and how they can influence known attacker behaviors. To actually keep it somewhat short, we primarily consider the very common, scalable, and continually devastating scenario where the attacker is remote.

Audience

If you have an interest in login security, authentication, or common mitigation practices, and find yourself looking for a general overview and evaluation, you've come to the right place.



Overview Of Attacker Behaviors

Attacker objectives, victims, and techniques vary significantly, even when just considering credential theft and misuse. We only scratch the surface here with common techniques in order to establish a foundation for later evaluation.

Objectives

Attacker objectives are as diverse as human ambition. As more sectors have moved to the internet for its many benefits, so too have attackers followed. Attackers include everyone from curious youth with nascent ethics, to organized criminals, to companies, to governments, and thus objectives range from pranks, to sowing chaos, to stealing money, to sabotaging elections, to stealing secrets for a leg up on the competition in business or in war.

Determined attackers tenaciously organize their resources and activities to achieve their objectives, and attackers across the globe with different objectives and skill sets often collaborate because the pseudo-anonymity and disintegration of distance on the internet makes it easy to do.

This can make it difficult to even understand attacker objectives, let alone attribute observed activities to particular actors or motives, though there are exceptions.

Victims

Anyone who can influence decisions, knows or has something, has access to something, or can contact someone who does, is a potential victim.

Some attackers cast wide nets and conduct low-cost activities with low individual returns, such as sending scammy ads for diet pills or pumping and dumping penny stocks. Step one rung up the investment/return ladder and there's inheritance fraud and stranded traveler scams.

Sophisticated attackers have specific victims in mind (individuals, industries, governments, labs, companies, bank accounts, intelligence agencies, public infrastructure such as electricity and water), and either target them directly with highly relevant and very hard to notice techniques, or sometimes buy access to those bigger fish from the others who happened to catch a few victims in a wider net.

Techniques

Attackers want to use the most reliable, least expensive, and least risky means to accomplish their goals. Showing up in person from halfway around the globe to break into something is both expensive and risky. Deploying weaponized custom exploits and payloads to the victim's devices and environments is also expensive and difficult.

Because stealing someone's password or other authentication data is relatively easy to do from afar, and there's little risk of or danger in getting caught, it's become one of the most common attacks in the world.

The remainder of this section will focus on common ways attackers obtain and misuse those credentials.

Weak Password Guessing

People aren't equipped or educated to create or remember good passwords. Sites also ask users to set up passwords before the account is valuable to them. Understandably, people often choose weak passwords out of ignorance, convenience, or because they don't think it matters, and rarely change them if circumstances change. This happens a lot. Attackers try common passwords with specific or common usernames across many sites all the time, and this can be surprisingly successful.

Password Reuse Abuse

It's also very common for people to use the same good or bad password, or a variant, across multiple sites. Attackers regularly steal credentials from one site and try them on another.

This problem is exacerbated by sites whose user information and credential databases are leaked, with potentially hundreds of millions of credentials available to attackers. There's even regular confusion if a collection of valid username/password combinations for a given site was due to a breach at that site or if it's collected from some other site or sites.

Attackers have also reportedly targeted weaker sites used by a given victim to gain credentials to try on the site they're actually interested in.

Man in the Middle (MitM) Attacks

Sometimes attackers have access to the network path between their victim's computer and the site they are accessing. This can enable the attacker to view what sites someone is accessing, or steal their data if the connection is not encrypted or if the victim is tricked into believing the attacker's system is legitimate. Attackers have even created this situation to achieve specific goals.

This privileged position can be used to wait for users to access the site of interest, or it can be used in combination with other techniques to entice someone to visit the site of interest.

Sometimes these attacks are noticeable by a skilled user (wrong URL, looks funny, no HTTPS, etc) and sometimes they are not (compromised certificate authority, homoglyph attacks, perfect site copies/relays, etc), but the vast majority of people probably won't notice–particularly if they are under a real or attacker-simulated deadline from their boss, spouse, the tax man, or similar.

Credential Phishing

Credential phishing typically uses some pretext to convince a person to reveal their credentials directly, or to visit some site that does the same. Attackers do this via SMS, email, telephone, instant message, social networks, dating sites, physical mail, or any other means they can. Phishing alone is responsible for huge monetary losses each year. Worse, many people aren't equipped to recognize when their credentials are being, or have been, phished.

Account Recovery Exploitation

Because of the large scale of many applications and the general desire to keep support costs low everywhere, account recovery flows can be much weaker than the primary authentication channel. For example, it's common for companies deploying strong two-factor authentication (2FA) solutions to leave SMS or OTP as a backup, or allow help desk personnel to reset credentials or set temporary bypass codes with just a phone call and little to no identity proofing requirements.

Attackers don't care if it's the primary authentication method or the recovery method–they'll take the easiest way in even if the sign on the door says "Members Only". The important thing for defenders is to strengthen both the primary and the recovery login flow so that users aren't compromised by the weaker path nobody is supposed to use.

Common Credential Misuse Mitigations

Once obtained, attackers can use stolen credentials immediately or at a later time, and often from anywhere. This is particularly problematic due to the complexities of modern account security options. People may be unable to fully secure their accounts post-compromise without help from an expert due to things such as email forwards, OAuth tokens, application specific passwords, and similar trust delegation–users may think a simple password change is enough, but without undoing these changes the account may remain accessible to the attacker.

Many defenders use techniques to make abusing stolen credentials more difficult. Below we quickly define and explain a few of the most common ones. The most effective strategies involve 2FA, and we spend most of our time on that area for that reason.

Be Hostile to Machines (but hopefully not to humans)

CAPTCHA, site obfuscation, rate limiting, and client fingerprinting (browser v/s script, positive karma, etc.) can make it harder, but certainly not impossible for attackers to make use of stolen credentials. Think of these as speed bumps and cones rather than roadblocks–often necessary, but not sufficient to stop abuse on their own.

Visual Trustworthiness Cues

This class of mitigations tries to show the user something they can use to determine the provenance or trustworthiness of a site. These techniques are getting better over time thanks to teams doing security focused user experience research. Sadly, most people just don't look at many of these indicators, and some attacks can be indistinguishable even for experts.

Below are some of the commonly used indicators, but none have proven strong in the face of real world attacks.

Browser Address Bar, "Secure" Designation, EV Certs

Various visual indicators are available to those with the skills and patience to check them every time, and who never make a mistake. "https", "Secure", or the green treatment given to the title bar for EV certificates can help, but a quick survey of what "Secure" beside https://badguy.com means amongst family or colleagues makes the weaknesses here clear.

SiteKey

Some websites allow the user to set a persistent picture or pattern for a given browser that the user is more likely to notice than the URL bar.

Users just didn't pay attention, and this is now being removed from most sites in favor stronger techniques.

Trust Seals

Some sites pay for third party assessments and display a trust seal provided by the assessor. Some also believe that users will look for these seals prior to entering their credentials, or that evil sites won't display those seals so users can't be tricked. The effectiveness of modern phishing campaigns with or without trust seals shows this is not a generally effective strategy.

Password Managers

Password managers help users create strong, unique passwords per site and, if integrated into the browser, most try to ensure they are given out only to the site at which they were originally created.That's a lot better than reusing the same password or variation everywhere, but there are some caveats: Browser integration has had many, many problems over time. Users override the site binding protections. If not handled with extreme care, cloud synchronization of passwords can expose all passwords at once if breached. Users often forget their master password, and the learning curve prevents most users from taking advantage of all features and protections consistently.

Password managers can certainly help mitigate attacker techniques such as weak password guessing or password reuse abuse, but it's no panacea.

Two-factor Authentication (2FA)/Two-step Verification (2SV)

The use of two-factor authentication or two-step verification is one of the most powerful techniques for strengthening credentials. The core idea is to require more than just a username and password when the risk is high, like when signing in from a new device or a new place, when other suspicious activity has been detected from that account or location, or when performing a risky or sensitive transaction.

For the sake of brevity, we avoid distinguishing between 2FA and 2SV or being pedantic about 2FA requiring two of something you have, know, or are. Instead, we focus on things commonly used for credential strengthening under either of these umbrellas. We also ignore common deployment weaknesses such as allowing account recovery using just username and password, or via a phone call or call back.

What follows is a list of commonly used 2SV/2FA methods and a brief assessment of some of their strengths and weaknesses as they relate to attacker behavior. One benefit of all of these methods is that when used only in unusual circumstances they might all cause the user to think twice. Particular attention is paid to phishing because it is such a widespread and effective tactic.

Knowledge Based Authentication (KBA)

You've seen these questions before: What's your mother's maiden name? Which of the following streets have you (not) lived on? What was your second pet's name? Your highschool mascot? There are even dynamic versions of these questions for those in a position to know, such as "did you go to the grocery store on Thursday?" or "where was your last vacation?"

Mostly these are a form of second password–something "secret" you know. In the worst case, the information is publicly available with predictable results, and in the best case, users will be phished for it along with their password.

Worse, the set of questions is often in widespread use, giving applicability across different sites.

Because the answers are sometimes considered less secret than passwords, and because providers want to be able to use fuzzy matching due to typos and capitalization, the verification information is often stored as plain text answers or reversibly encrypted even when password verifiers are stored in better ways.

OTP or Notification via SMS or Phone Call

OTP via SMS has similar benefits and drawbacks to OTP tokens and apps, but there are some significant differences. On the positive side, there are no secret seeds to be stolen, and most people already have a phone number. Also, because access to someone's phone number is a matter completely separate from internet services, it's particularly seductive from a support perspective–if you have access to a phone number, you're a particular individual.

Sadly, it gets quite a bit worse from there. The user must give up their phone number, which can be considered private information. Phone and SMS networks have been exploited by governments, criminal gangs, and even penetration testing firms. Phone companies can provide access if required to by legal order or if hacked or tricked.

Relying parties also necessarily trust all account validation and recovery methods provided through the user's phone provider. For example, pretexting/vishing, number porting fraud, or even using false identity documents in phone company stores are weak points.

And of course it's still vulnerable to phishing just like other OTP mechanisms.

OTP or Notification via Email

OTP via email instead of SMS changes things a bit, but not by much. There's still the outsourced support temptation. The user already has an email address, but there are new attacks too: BGP hijacking can let attackers see email traffic containing password reset links or OTP email codes in real time. Email is remotely accessible by design, so if the attacker gets access to someone's email, they likely get access to everything else too. This also means the relying party necessarily trusts all authentication and account recovery methods the user's email provider supports. That might not be a huge problem for major email providers, but not every email provider is created equal.

The BGP hijacking issue might get better as email security improves with STARTTLS STS, DANE, and similar encryption technologies, but ultimately email is a slow and cumbersome way to provide account security that doesn't stop phishing.

One Time Password (OTP) Tokens and Apps (OATH TOTP/HOTP, etc.)

The core innovation of OTP tokens and apps is to give the user a secret seed–typically by embedding it in a hardware token or having the user capture it using a QR code–and using that seed combined with a counter or the current time to produce a code that is unpredictable without knowing the seed. An OTP code can not be used to recover the secret seed, and is only valid once. The user doesn't know the seed so they can't be phished for it, and if they're phished for one OTP code, it can only be used once, and must be used in near real time. This is a huge step up from static KBA questions.

There are also a few weaknesses. First, the secret seeds must be present on a server somewhere in order to validate the OTP codes, which means they can be stolen unless extreme care is taken. Second, many manufacturers keep the seeds so they can help their customers recover, and some have been catastrophically breached. Finally, users still give out OTP codes to phishing attacks, so attackers don't necessarily have to change their strategy away from phishing, they merely have to tweak their tactics to capture and use the OTP in near real time.

Push Notification-based OTP codes

When implemented correctly, OTP via push notification doesn't have the same interception issues as SMS or email, and even in simple implementations only push notification providers such as Apple and Google can intercept them. As with all OTP implementations, however, the user will simply give the code to the attacker when phished.

Push Notification-based Approval Apps

These applications provide the user with some context–e.g. "you're logging into your bank from El Paso"–and an approve/deny button you touch instead of a code to enter. When implemented well, these apps have all the benefits of a good push OTP implementation, but are more usable.

Additionally, because application information can be included in the message sent to the user, an attacker must change their phishing tactics to not only use stolen credentials in real time like with OTP, but must also phish the user for the specific app if the user reads the message carefully.

Attackers may also need to use a bot (infected computer) with a similar location or ISP to the user's device to avoid suspicion if the app displays the user's geolocation, and if the user reads the approval message carefully. When phishing a user, attackers have access to the user's internet location in addition to their username and password, and use that information to control the message
the user sees.

When a user receives and approves a login approval push notification when being phished, it can actually reinforce the belief that credentials were entered into a legitimate site–especially when the app and location match what the user expects. This is a weakness for all solutions that rely on the user to know for sure if they are accessing a legitimate site or not.

Biometrics

Biometrics use something intrinsic to a person's body to augment or replace other mechanisms in the authentication process. Some of the most common examples are fingerprints, iris scans, or facial geometry, and modern implementations can be quite easy to use. Unlike other authentication credentials which can be changed if they are exposed, you can't get new fingerprints or a new face if yours become publicly known.

Over time, many improvements have been made to prevent forged biometric identifiers from being used, but today, widespread biometric sensors are quite frequently fooled by a local attacker with no specialized equipment. Because biometric implementations are complex and varied, users sometimes perceive them as magically secure from attackers and can be less likely to be skeptical of, for example, placing their finger to perform some action. However, unless the underlying authentication protocol used between the device and the relying party has phishing and MitM protection, the magic is only skin deep, and users or even experts can't trivially tell the good from the bad.

While there are systems that use central validation of biometric data, the privacy implications of those systems are obvious, and they are rare compared to device local biometrics. Biometrics are most often enrolled and validated local to a specific device, and used to unlock a secret which is in turn used to authenticate the user to a service via a non biometric protocol. There is typically no way to move credentials between devices, and sensors are attached to devices that are fragile, and expensive, and are replaced regularly. This means re-enrollment is often frequent, and the security of not just the underlying authentication protocol but also the re-enrollment process are key.

Despite this, when validated locally to a device, if the underlying protocols are chosen and implemented carefully, and if the account recovery and rebinding mechanisms are attacker resistant, they can prevent scalable remote attacks.

Manufacturer-provided biometrics on mobile devices can be shared between relying parties' native mobile applications without requiring re-enrollment, but web use is not widely available today and desktop/mobile disparity can prevent widespread adoption.

Most implementations require a password or PIN before allowing biometric sensors to be enrolled, and in most of these cases that PIN can be used to access all information and secrets on the device. This is usually less of a concern in the case of a remote attacker, but might need to be considered depending on the threat model. One can't assume that a phishing and MitM resistant software application using a biometric sensor is significantly different than an app using a PIN unless extreme care is used in the implementation.

Finally, there is the issue of legal differences between compelled identification (fingerprint/iris) v/s compelled testimony (password/PIN). This is usually far more serious an issue for unlocking keys or encryption than for authentication, but is increasingly interesting for both while traveling.

Certificates

Certificate-based authentication, occasionally called "TLS mutual authentication", uses public/private key cryptography. The user has a private key and certificate, and proves possession of the private key in order to authenticate to a server.

Certificates are already used to validate the server side of every https connection, and enterprises have used certificates to authenticate their internal users for years–both with and without smartcards to protect the private keys.

Authenticating users with certificates has the benefit of preventing both phishing and MitM attacks. This is because the TLS protocol ensures the server can detect if there is something interfering with connection between the server and the client.

Fortunately, in enterprise environments, user certificates with smart cards can be both usable and secure. A few internet websites–mostly banks–have used user certificate-based authentication, but without careful management, certificate-based user authentication systems can be quite hard to use and manage. For widespread internet use, something different is generally needed.

FIDO Universal 2nd Factor (U2F)

When creating FIDO U2F, Google and Yubico wanted the security and phishing/MitM resistance of smartcard-protected keys and certificates, but wanted it to be highly usable on the web and privacy preserving.

FIDO U2F authenticators use public/private key technology to create a unique key per registration and site to preserve privacy, use dedicated secure hardware for handling sensitive private keys, bind credential use to only the site at which the credential was created, and require user interaction to authenticate. Servers can also detect and stop MitM attacks by using Channel/Token Binding.

FIDO U2F's phishing resistance requires the underlying operating system/browser to accurately represent the URI of the site to the authenticator, and that is guaranteed unless the browser or operating system is already compromised. Most importantly the protocol does not require the user to notice or confirm anything about the site they are visiting to stay safe–it "just works." Also, because authenticators are separate devices, there is far less need to go through cumbersome account recovery regularly. This allows recovery to be strengthened without significantly impacting users, and can therefore prevent attackers from exploiting account recovery weaknesses.

Chrome has natively supported FIDO U2F authenticators since version 38. Firefox has had support via plugin since late 2015 and will add native support soon. Microsoft has announced that they are working on native support in Edge, and Opera 40 and later versions already have native support.

U2F is easy for both enterprises and internet providers to use, and has been deployed by Google, Facebook, Dropbox, GitHub, and many other very large providers. Google even published a paper showing how their internal deployment increased security while decreasing support costs and authentication time.

Summary

Anyone is a potential victim on the internet, and the stakes are higher than ever. Today, in many environments, attackers can still guess common passwords, reuse them from site breaches, and phish credentials and use them with impunity. They occasionally have to tweak their tactics to deal with password managers, rate limits, SMS, or push approvals, but their playbook is still mostly the same unless stronger mitigations are used.

Education can only go so far given the realities of how services are used and the way people do their work. Even the head of the social engineering CTF at Defcon can be fooled. Sites should provide–and users should demand access to–the best technology available to protect from credential theft and subsequent abuse.

If you are building or maintaining a service, we hope that you will use the information provided here to help steer your design, and if you're a user, you should now better understand how easy it is for all of us to be fooled and how technology can help.

yubico

About Yubico

Yubico sets new global standards for simple and secure access to computers, mobile devices, servers, and internet accounts.

The company's core invention, the YubiKey, delivers strong hardware protection, with a simple touch, across any number of IT systems and online services. The YubiHSM, Yubico's ultra-portable hardware security module, protects sensitive data stored in servers.

Yubico is a leading contributor to the FIDO2, WebAuthn, and FIDO Universal 2nd Factor open authentication standards, and the company's technology is deployed and loved by 9 of the top 10 internet brands and by millions of users in 160 countries.

Founded in 2007, Yubico is privately held, with offices in Sweden, UK, Germany, USA, Australia, and Singapore. For more information: www.yubico.com.