IMMUNIO

Account Takeover

How Hacking Happens Today

authored by Mike Milner
for **IMMUNIO**

IMMUNIO /eBooks

Account Takeover

How Hacking Happens Today

Hacking happens. It happens to the best websites and networks. However, by understanding a little bit of what is inside the head of those who want to hack your site and the techniques they use, you can improve your security posture and reduce the likelihood of a breach.

authored by Mike Milner
for **IMMUNIO**

Co-founder and CTO
@secretmike

Hacking happens. It happens to the best websites and networks. However, by understanding a little bit of what is inside the head of those who want to hack your site and the techniques they use, you can improve your security posture and reduce the likelihood of a breach.

There are two main ways that hackers get into your website. They either look for a technical vulnerability to exploit, like a bug in the code, or they get the credentials of someone with access, and waltz right in like they own the place.

Finding exploitable vulnerabilities in a website is not rare, but it usually requires specialist skills, and needs to be repeated for each website being attacked. Increasingly attackers are aiming at a much softer target—taking over the accounts of your users.

Account Takeover (ATO)

There are a few key tricks up the sleeves of today's hackers:

- Finding passwords to reuse from credentials leaked on another site (credential stuffing),
- Brute forcing weak passwords,
- Phishing and social engineering schemes,
- Session stealing through a code vulnerability or lack of encryption.

All of these target the accounts of your users, and they can be used effectively against almost any website. These techniques are becoming so easy in fact, that Verizon Data Breach Investigations Report (VDBIR)[1] has found that the use of stolen credentials has been the number one attack vector for web applications over the past two years.

[1] *2016 Data Breach Investigations Report, Verizon Enterprise*

BREACHES

LinkedIn
In 2012, **6.5 million user names and passwords stolen**. In 2016, reported by Data Breach Today to be as many as 117 million.[2]

Target
In 2013 **40 million accounts hacked**; the hackers got credit card information.[3]

Google
In 2014, **5 million gmail usernames and passwords** stolen, probably by phishing, botnets, hacks of third-party sites.[4]

[2]*http://www.databreachtoday.com/linkedin-breach-worse-than-advertised-a-9113*
[3]*http://money.cnn.com/2013/12/22/news/companies/target-credit-card-hack/*
[4]*http://www.databreachtoday.com/google-locks-down-stolen-credentials-a-7303*

CREDENTIAL STUFFING

Hackers use existing data to take advantage of the fact that we humans are the weakest link when it comes to digital security. Despite numerous admonitions to the contrary, up to 50% of us use the same usernames and passwords across sites.

Criminals like the fact that users hate passwords. And that users often reuse their credentials across websites, repeating usernames and passwords. This makes it easier for the user to access websites. And it makes it easier for the hacker to profit from stolen credentials.

At its simplest form, hackers take stolen usernames and passwords from other breaches, and stuff them into your website to see if any of them work. If they do, they're in. If not, it wasn't much effort to write the script in the first place. The industry calls this practice credential stuffing.

According to the Verizon report, 63% of confirmed breaches in 2015 involved "weak, default, or stolen passwords." And the most common threat action associated with attacks involving legitimate credentials was: Hacking—use of stolen credentials, with 1095 incidents.

The thing about leaked passwords is that once they get out there, they are out there for as long as they are useful. There are websites that aggregate leaked passwords, technically for the use of consumers to find out if they've been compromised. Of course the hackers can see these sites, too. This is becoming an easy source for leaked credentials. Take pwnedlist.com—this site hosts 866 million records from public password breaches. They recently discovered (and fixed) a vulnerability that potentially exposed all those records to hackers. As consumers get more savvy and update credentials regularly or as a result of a breach, the dream is that these lists will become ever less useful for those who seek to exploit them.

TOP 10 WEAKEST PASSWORDS 2015[6]

1. *123456*
2. *password*
3. *12345678*
4. *qwerty*
5. *12345*
6. *123456789*
7. *football*
8. *1234*
9. *1234567*
10. *baseball*

BRUTE FORCE OF WEAK PASSWORDS

It's bad enough that end users use the same credentials across websites. But they are also prone to creating easy to remember passwords, which usually makes them quite weak in terms of their ability to be guessed. So even if a user doesn't reuse credentials across sites, a weak password threatens a system's security.

This is especially bad because it is sometimes possible to get from a website itself all the user names used on that site. If the site is not secured to protect against that, a hacker can get a list of valid usernames. They will then enter the valid user names and check the password 123456. Statistically, they will access about 5% of the accounts on the site this way.[5]

For companies, the defense against weak passwords is rate limiting. And watching the site—chances are if someone is trying the password 123456 with a user name, and the password is not valid, they are testing the system. Any system that recognizes the testing of weak passwords, like 123456, can then shut down this user sooner rather than later.

[6] *"Announcing Our Worst Passwords of 2015," Splash Data, https://www.teamsid.com/worst-passwords-2015/*

[5] *Coles, Cameron, "You Won't Believe the 20 Most Popular Cloud Service Passwords," Skyhigh Networks, https://www.skyhighnetworks.com/cloud-security-blog/you-wont-believe-the-20-most-popular-cloud-service-passwords/*

The median time for the first user targeted by a phishing campaign to click on the attachment was 3 minutes and 45 seconds.

PHISHING

Hackers still reach right out and digitally touch you. And they really want you to touch them back. Targeted attacks take advantage of one of the very qualities that make us humans tolerable to live with—trust in others. Trust, that is, of either businesses or individuals we know and love.

They like to phish, a form of social engineering to trick users into giving up their information. These hackers (89% of which are organized crime syndicates, Verizon reports) send emails that look like they are from a legitimate business an end user knows and trusts, like the bank. Perhaps saying that the account has been hacked. And providing a helpful link to the site, so the user can easily resolve the issue. The link takes you to the hacker's site, that looks so much like the real site you cannot tell the difference. Then they steal the user name and password once it's entered.

Phishing works—Verizon[7] reports that 30% of phishing messages were opened by the target across all campaigns and about 12% clicked on an attachment to "enable the attack to succeed." If you doubt it, Verizon also notes that the median time for the first user targeted by a phishing campaign to click on the attachment was 3 minutes and 45 seconds.

Companies are frequently reminding users not to click on links in emails like this, to instead go directly to the site by typing in the trusted URL. And tell their users that they will never call or email you asking for a username or password. Yet, people still do fall for these tricks; in enough numbers that it is worth the hackers' time.

[7] *2016 Data Breach Investigations Report, Verizon Enterprise*

CODE VULNERABILITIES

There are some hacks that do not focus on the foibles of human behavior, but rather look for flaws in the machine. Session stealing is when a legitimate user logs in to a site, and someone steals the session to get access to that account.

Each time a user accesses a website, they are using a session—the system passes a secret to the web browser the account holder is using, to prove that he or she is a legitimate user with an account. When the user logs out, the session ends and that's that. Even the back button shouldn't start the session again. It is frustrating for users when this happens, as they have to log in again if they want to access the site again. But it's for their own protection.

If the website is vulnerable, it is possible for someone to steal the session. If the user is accessing the site over unencrypted wi-fi and a nefarious individual, such as the man at the next table in the coffee shop, is listening in, he can steal the session and the site would think it is still a valid session with the legitimate user. In this case, the attacker doesn't get the account password, but he or she can get other details and data.

Preventing Account Takeover

Taking over accounts is all about volume. Attackers have access to hundreds of millions of potentially active credentials. They need to test as many as they can, as fast as they can. This means bots.

Credential stuffing and brute force are so common now that there is dedicated bot software for running attacks. Even amateur hackers, or script kiddies can use tools like Sentry MBA—a stand-alone, Windows application—to launch sophisticated distributed attacks.

It's not even necessary to manage your own botnet these days—you can just rent one. Now, instead of the website having to defend against one pseudo-browser sending hundreds of requests, it must deal with hundreds of fake users sending one or two requests each. Maybe spread out over time, with each compromised computer checking only a couple of passwords each day.

SENTRY MBA ATTACKS[8]

5 million+

login attempts in one week, using hundreds of thousands of proxies, at a Fortune 100 B2C website (Dec 2015)

30,000

login attempts of company's website application and 10,000 login attempts of company's mobile APIs (Dec 2015)

20,000+

login attempts in two attacks over two days at a large retailer (Jan 2016)

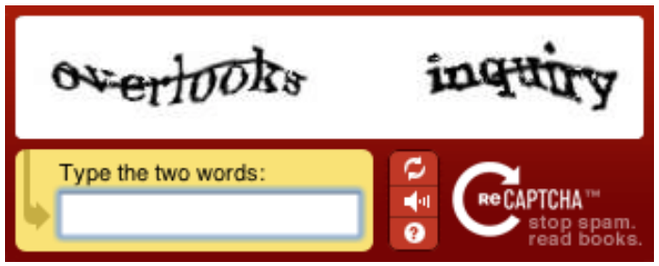
10,000+

login attempts, using over 1,000 proxies, at a large retailer (Jan 2016)

[8] *Agarwal, Sumit, "The Half-Day Attack: From Compromise to Cash with Sentry MBA," Shape Security Blog, 9 March 2016, https://blog.shapesecurity.com/tag/credential-stuffing/*

THE CAPTCHA

The basic tool used to stop bots is a CAPTCHA. A CAPTCHA is a challenge designed to be as easy as possible for a human to solve, while at the same time being very difficult for a computer to solve. A small self-contained Turing test.



A CAPTCHA is designed to be solvable by a human, but they're still very annoying. You could stop bots by asking everyone to solve a CAPTCHA as part of logging in, but it makes the process very frustrating for your users. The trick is to be selective—show the CAPTCHA when you see a bot, don't show it when you see a human. Attackers know this, so they try to make their bots look just like a human user. As a defender you need to focus on telling the difference.

BASIC DEFENSE

Simple bots can be identified just by looking at details of a request. Each HTTP request contains headers from the requestor. These headers may contain clues that help tell the difference between humans and bots. A common one is the User-Agent header, which identifies the software making the request. For web browsers this contains the browser name, version, operating system, etc. Bots usually spoof this value to make themselves look like a browser, but it's possible that they make mistakes. If you can spot these mistakes it's very easy to stop these simple bots.

There may also be an indicator in the timing. Humans usually pause to read a page before moving onto the next page. A simple bot may have a constant rate of requests that will make it stand out from humans. Most bots incorporate some randomness to make the requests look more natural however.

If the bot is a bit more sophisticated, it is generating requests that look exactly like a

real browser, and spreading them out to look a bit more natural. The attacker still wants to try lots of password combinations, so if the bot is making a huge number of requests in a short timeframe, you can usually identify it just by setting thresholds on your login forms. Too many attempts from an IP address indicate a bot.

The most sophisticated bots take things one step further, and spread their requests across a zombie army of computers called a botnet. A botnet is a collection of infected computers, usually spread across the globe. Botnets are easily available for rent, and allow a single attacker to launch an attack from hundreds and sometimes thousands of IP addresses all over the world. This makes rate-limiting techniques effectively useless. Each source IP address may only try one or two invalid login attempts each day—well below the threshold of a normal user who just can't remember which password they used.

For these sophisticated bots, detection requires some additional techniques: Threat intelligence, and client-side profiling.

THREAT INTELLIGENCE

The basic idea of threat intelligence is that a botnet will be used to do many bad things by different attackers. Threat intelligence providers collect lots of data about attacks and aggregate the data to make a list of known-bad IP addresses. Your job is to get the list and keep it up to date. If a request comes from an IP address on the list, you assume it's a bot and show the CAPTCHA. This lets you stop a bot from the very first request.

The difficulty with threat intelligence is finding a good source and integrating it into your systems. You could potentially block IPs that are on the list, but you're putting yourself at the mercy of the threat intel provider. If they accidentally add Google to the list, you may find your search engine rankings decline dramatically. With careful integration you can set up a process to evaluate the effectiveness of each threat intelligence feed.

It's also very important to keep the list up to date. Many providers update their lists continuously as new IPs are added and old ones removed. If you let your list fall out of date, you may be blocking the wrong IPs.

It's also possible to contribute data back to a threat intelligence source. By pooling evidence-based knowledge of existing threats and breaches and sharing it in real time, companies gain critical knowledge about how to detect and block threats to their own system.

CLIENT-SIDE PROFILING

Sophisticated bots make requests exactly like real browsers. If a botnet isn't part of a threat intelligence feed, a defender has little chance detecting a bot based on the network requests. For cases like this you need to reach further, to profile the client itself.

Real web browsers are very complex pieces of software. When they load web pages, they process the result in very specific ways: they handle cookies, parse and render HTML, load CSS files and image files, execute javascript. By embedding specially crafted content in a page, it's possible to observe the behavior of the browser. If some images aren't loaded, or certain javascript doesn't run, it's a strong indication that the client is not a real browser.

If an attacker is using a headless browser like PhantomJS it will execute javascript like a browser, but it's possible to then use the javascript to further fingerprint the environment, to distinguish between a real browser and a scriptable headless browser.

This goes back and forth—the defender makes a new check to detect bots, the attacker updates the bot to fake it so it looks even more like a real browser. Proper defense requires continuous monitoring and enhancement.

USER:
Mike Jones

DEVICES

OSX Safari

Android Chrome

LOCATIONS

1.2.3.4 California

2.3.4.5 Montreal

Detecting Compromised Accounts

Even with the best prevention techniques, it's still possible for your customers' accounts to be taken over. Phishing attacks can compromise an account without hitting your real website at all. Code vulnerabilities can be used to steal access to a session in progress. For these reasons you need a strategy to protect accounts that have already been compromised.

The most common technique is to maintain a profile for every one of your users. Details on what type of devices they use to access your site, and when they typically use it can both help flag when some behavior has changed. It's also possible to keep track of where geographically a user accesses the site. If a user always logs in from New York City, then suddenly logs in from China, that's a strong indicator that the connection from the new location is trying to use a compromised account.

For review or exchange sites, you know how your typical user behaves and can guess how a spammer behaves. So if someone logs in and immediately posts an ad when that is not their usual MO, you'll be able to flag that account. Then the system should send a CAPTCHA or ask a security question to prove it's a real person.

Criminals work hard to exploit whatever vulnerabilities they can find in your website and network. And they make a lot of money doing it.

Conclusion

As you can see, these criminals are dedicated bunch. They work hard to exploit whatever vulnerabilities they can find in your website and network. And they make a lot of money doing it.

On the side of the defenders against these attacks, it's an ever escalating arms race. But know that a little bit of the right security goes a long way. These are some of the most basic threats and most basic steps to take to prevent hacking via stolen credentials and to prevent a great loss for your company.