January 2021

Top 10 Hacks of the Decade

Teleport

Top 10 Hacks of the Decade

In 1971, when employees booted up their PDP-10 mainframe computers, they found a peculiar message waiting for them at the teletype console.

"I'm the creeper: catch me if you can."

This wasn't a single instance of a prankster printing something off; the mysterious message was found at multiple stations throughout the BBN Technologies headquarters. What researchers were witnessing was the spread of the first computer virus, a worm called Creeper. At the time, Creeper was not considered a virus or a worm. That type of language was relegated to the hypothetical realm. No such program had ever existed before.

Half a century later, there's a dictionary full of terms used to describe such programs: Worms, boot infectors, Trojan horses, malware, spyware, etc. From its humble beginnings out of an office in Cambridge, Massachusetts, cybersecurity has become a dominant pillar of any organization. Whether it's IBM with an army of security professionals or a mom-and-pop store using Gmail, it's impossible to dream up a world in which anyone would leave their data wilfully exposed.

Cybersecurity as an industry is reactive to external threats. Practices evolve on the backs of hacks that change the way professionals think about security. This whitepapers takes a look at 10 of those hacks, particularly ones that occurred from 2010 onwards.

Operation Aurora (2010)

What was Operation Aurora?

The first hack on the list began slightly before 2010, but it's public acknowledgement in January kicked 2010 off with a massive cyber attack, titled Operation Aurora. Operation Aurora is mostly associated with Google, who announced that they had been on the receiving end of a sophisticated cyberattack which was later traced back to a foreign government. However, the attacks had a much broader scope, having targeted other organizations like Adobe, Juniper Networks, Dow Chemical, Morgan Stanley, and a dozen others. The primary goal of Operation Aurora was to steal intellectual property in the form of source code, and in many cases, succeeded.

Operation Aurora changed the way that cybersecurity professionals viewed threat models. Attacks against corporate networks were nothing new. However, most cyberattacks were relatively unsophisticated, drawing on the restrained resources and expertise of small groups of criminals. Operation Aurora was created and operated by a nation state with the resources to not cut corners and fully obfuscate their actions.

How did it occur?

Operation Aurora targeted a zero-day exploit in Internet Explorer as its method of entry into over 30 companies. To get access to a browser that ran on corporate networks, operators employed a spear phishing campaign, targeting a selected group of users. These users received plausibly trusted messages, either through email or chat apps, that directed them to a site based in Taiwan. Once users clicked the link, a JavaScript program ran in-browser, exploiting the Internet Explorer zero-day, allowing the browser to download an encrypted file that contained the malicious software. This software opened a backdoor on the computer, allowing operators to remotely access the computer and search internal networks for credentials, IP, or anything else that computer was authorized to access.

Much of the theft perpetrated by Operation Aurora remains private, but investigations have confirmed that, in some cases, project source code was stolen. In Google's case, code for their prototypical-SSO program, known as Gaia, was exfiltrated along with Gmail data from political dissidents.

What happened afterwards?

Aside from the public cyberespionage of corporate networks and theft of IP, Operation Aurora forced Google to reconsider their approach to securing their networks. Four years later, Google published a research paper titled *BeyondCorp: A New Approach to Enterprise Security*. In it, they detailed a new type of security architecture that did away with VPNs and relied on a modern set of assumptions about how networks work and the threats they face.

This method of security had been mulling about for some years at this point. Formally known as “Zero Trust,” Google was the first company to implement these principles at scale. In the years following, cybersecurity professionals have followed Google’s lead and sought to re-architect their infrastructure and modernize enterprise security practices.

Stuxnet (2010)

What was Stuxnet?

Just as the Creeper virus represented a once-theoretical program manifested in reality, so too was Stuxnet. Stuxnet, uncovered in 2010, was a malicious computer worm, designed to target supervisory control and data acquisition systems used for control industrial processes. In this case, the Stuxnet virus proliferated through machines at the Natanz plant in Iran.

Stuxnet was deliberately designed to target a specific set of machines that met certain configuration requirements, leaving the virus inert in those locations. In fact, Stuxnet was never meant to spread beyond its initial target, as the Natanz power plant was air-gapped and not connected to the internet. However, due to an error during a software update, the worm did find its way onto the internet, where it extended beyond anyone’s control.

How did it occur?

Stuxnet was a complex virus without precedent. It consisted of three main modules:

- The worm that executed the Stuxnet payload
- A .lnk file to execute copies of the worm
- A rootkit to hide itself from malware detection and human operators

In addition, Stuxnet had a number of other complex features, including a command and control network that allowed its creators to remotely access it, provide commands, and update the program.

Stuxnet used a number of zero-day exploits to spread across Iran's nuclear facilities. However, the initial infection method was via USB, as Iran's nuclear facilities were not connected to the internet. The virus's creators originally targeted contractors that were thought to be working with Iran. Having infected their computers, the virus was able to spread through local networks via a number of vulnerabilities including hard-coded passwords, shared files, and zero-day exploits.

Once the worm successfully infiltrated the plant's industrial control systems, it sought out programmable logic controllers (PLC) responsible for issuing commands to the centrifuges that enriched Uranium. Once hijacked, Stuxnet issued commands to the PLC that manipulated centrifuges, eventually damaging them until they ceased to function, all the while relaying falsified information to the end user to indicate the centrifuges were running normally.

What happened afterwards?

Stuxnet's largest impact was not the damage it did to Iran's nuclear power plants or the political fallout afterwards. Instead, Stuxnet legitimized cyberspace as fair game to weaponize. Stuxnet was not the first attempt at cyberwarfare, but it was the first successful attack on physical infrastructure. The novel nature of the attack, as well as the highly publicized discovery and investigation signaled to the world that such attacks were fair game, effectively kicking off another arms race.

In the years since, the nature of cyber-attacks and cyber-defense has rapidly changed, leading to the development of cyber-crime, cyber-terrorism, cyber-espionage, and other attack vectors that have been grafted into cyberspace.

Mt. Gox (2014)

What was Mt. Gox?

Switching gears from mainstream hacks, Mt. Gox was the largest theft of Bitcoin to date. At the time, Mt. Gox was the largest Bitcoin exchanged in the world, accounting for over 70% of all Bitcoin trades. But in February 2014, the exchange suddenly halted all Bitcoin withdrawals, claiming to be investigating a technical issue. A few weeks later, the exchange suspended all trading and went offline. Internal documents had revealed roughly 850,000 Bitcoin had been stolen between the company and its customers.

At the time, the amount stolen totaled 450 million USD. As Bitcoin experiences another boom, that comes out to over 34.5 *billion* USD in January 2021. Of this, 200,000 Bitcoin was recovered, distributed to Mt. Gox's creditors after it filed for bankruptcy. Investigations since then are still ongoing, though the majority of the Bitcoin stolen cannot be traced.

How did it occur?

Unlike prior hacks, Mt. Gox was a fairly simple hack, primarily a result of a poorly managed codebase. The 2014 hack relates to its first hack in 2011, when 2,000 Bitcoin stolen from hundreds of wallets, used to store the cryptocurrency. The infiltrator gained privileged access to the exchange's software by stealing the credentials of an auditor that worked with Mt. Gox. With escalated privileges, the attacker modified source code to artificially reduce the value of Bitcoin, transferred 2,000 Bitcoin onto the exchange, and then sold them to him/herself.

The exchange responded professionally, crediting back 2,000 Bitcoin out of their own coffers and changing the storage method from a hot wallet (connected to the internet) to cold storage (not connected to the internet). In the years that followed, Mt. Gox's response to their 2011 incident proved to be ineffective as infiltrators still had access to private credentials and slowly siphoned Bitcoin while obscuring their transfers are normal transactions on the exchange. The dysfunctional operational security at Mt. Gox remained oblivious to leak until 2014, when the public was made aware of Mt. Gox's insolvency.

What happened afterwards?

The majority of Bitcoin stolen from Mt. Gox was never recovered, and likely never will be. In the public sphere, Mt. Gox was a testament to the juvenile nature of cryptocurrencies, decried as unsafe and a scam. Within the crypto-community, it sparked a debate about centralized exchanges that had ownership of the cryptocurrencies in wallets they controlled. An analogy to traditional enterprise markets would be trusting a third party to manage private company data.

Nowadays, centralized exchanges like Coinbase or Binance offer much better peace of mind for customers with transparent operations, scrutinized security, and insured deposits. But as with Mt. Gox, they are also targeted by competent hackers.

Panama Papers (2016)

What was the Panama Papers leak?

Next on the list is another example of terrible security practices. The Panama Papers refers to a leak of over 2.6TB of data, considered to be the largest leak in history. The data, obtained by a pseudonymous actor, John Doe consisted of 11.5 million documents, including emails, PDFs, images, and other file formats. Aside from being the largest incident of its kind, the Panama Papers leak made headlines as it exposed high-ranking officials from all over the world using offshore companies to hide income and avoid taxes.

How did it occur?

At the heart of the breach is the Panamanian law firm, Mossack Fonseca. Though they were responsible for storing sensitive information from some of the world's most influential people, Mossack Fonseca had appallingly poor security measures. Consider some of the many vulnerabilities later discovered by security researchers:

- Mossack Fonseca's customer portal, which ran an old version of the Drupal customer management system, had not been updated since August 2013. This version is known to be susceptible to at least 25 different vulnerabilities
- Similarly, Mossack Fonseca ran web servers using an outdated version of WordPress that was known to be susceptible to unauthorized access via its Revolution Slider plugin, giving the attacker shell access to the web server
- Emails were not encrypted over TLS
- Mossack Fonseca's web servers were not protected by a firewall and were on the same network as their mail servers

So how exactly was the data stolen? No one is certain for sure, but a number of different methods have been theorized, given the number of vulnerabilities Mossack Fonseca was exposed to.

What happened afterwards?

The Panama Papers hack provides a case study in some of the most basic cybersecurity principles: Keep software updated, segment networks to prevent lateral movement, encrypt using TLS, do not store passwords in plaintext, etc. Any professional worth their salt knows these precautions, but what is worth noting that the illegal data obtained in the hack was used as evidence in prosecution. Now more than ever, companies store sensitive customer data. The Panama Papers serves as an example for the potential fallout of unsecurely managed data. Suffice to say, Mossack Fonseca does not exist anymore.

The DNC Hack (2016)

What was the DNC hack?

In two separate events, the Democratic National Committee and Hillary Clinton's campaign in the United States were hacked, capturing the public's attention more than any other incident in 2016. A little over a month before the 2016 election, WikiLeaks published a trove of emails about Hillary Clinton and her campaign, obtained by the operative group, Fancy Bear. An act that would dominate the news cycle for the next month.

How did it occur?

The Clinton Campaign had quite a few measures in place to prevent hacks, including two factor authentication, regularly wiped email servers, and phishing drills. This kept the perpetrators at bay for a while, requiring multiple spear phishing attempts - the same method as Operation Aurora. After failing to infiltrate official campaign email accounts, attackers went after personal Gmail accounts, which were not held to the same standard of security as official accounts. On March 19th, hackers had successfully duped one of their targets into resetting his password and directed him to a fake website. With access to Podesta's email inbox, Fancy Bear extracted 50,000 emails.

At some point in their efforts, attackers had also obtained system admin credentials that had unfettered access to the DNC network. Having identified devices that were connected to the network, Fancy Bear installed malware on each device. The tools, X-Agent and X-Tunnel, allow them to log keystrokes on devices and extract data through a series of buffer servers to their eventual destination. Altogether, perpetrators had exfiltrated 300 gigabytes of data from the DNC.

What happened afterwards?

In the same way that Operation Aurora provided an overt reason to conduct cyberwarfare, the DNC hack had the same effect on elections. It became clear that attacks that undermined the integrity of the voting process was an inevitability. So to say the DNC Hack heightened the American government's response to cybersecurity threats would be an understatement.

In the years that followed, election officials poured over a billion US dollars into upgrading voting security infrastructure. Methods included moving away from highly vulnerable paper ballots to voting machines, upgraded voter databases, multi factor authentication, and security training for thousands of poll workers. These measures were put to the test in 2020 when COVID-19 turned the election registration process into a largely digital endeavor, but jurisdictions were able to stay online. As for the DNC itself, their updated measures included specialized hardware, curated cloud services, and of course, lots of phishing drills.

Equifax (2017)

What happened in the Equifax hack?

Equifax holds a treasure trove of valuable information in its servers. As one of the largest credit reporting agencies in the world, they stored sensitive personal and financial information on many Americans. What made the Equifax hack such a sensation was a combination of the breadth and depth of the data stolen. Breadth in that the hack is estimated to have affected 143 million Americans, or more than 40% of the country's population, and depth in that names, addresses, social security numbers, and drivers' license numbers were stolen.

How did it occur?

On March 6th 2020, Apache released a security notice identifying a vulnerability in Apache Struts, an open source framework for building Java web applications. The vulnerability, named CVE-2017-5638, allows code to be remotely injected through HTTP headers, which Struts could be tricked into executing. In their notice, Apache recommended that anyone using Apache Struts upgrade their version that had patched the vulnerability.

Equifax was quick to react, but along the way, something went wrong. Likely due to human error, the patch was not fixed at Equifax and the scans they ran to identify unpatched systems failed to report any vulnerabilities. In the meantime, the global hacking community had begun to run their own scans on the web, looking for places to exploit CVE-2017-5638, and Equifax was quickly identified due to their public-facing customer portal.

For months, a sophisticated group of attackers exfiltrated data for months, hopping from database to database in a largely unsegmented network. With massive amounts of data being extracted, Equifax surely would have noticed the odd behavior much earlier on. But having not renewed third party security software, the encrypted data flowing through Equifax's networks was not inspected.

What happened afterwards?

Unfortunately, the Equifax hack did not do much to change the status quo. In 2018, as more information became available, their earnings dipped, but quickly caught back up in 2019. The company invested $1.4 billion into security upgrades and another $1.4 billion in settling claims, which amounted to a measly $125 in compensation per customer affected.

Equifax is a case study in legacy companies that are slow to react to modern advancements. Despite spending millions on security tools, poor implementation and poorer governance led to basic failures like not updating software and renewing licenses. In addition, Equifax was unable to keep up with modern network practices, such as segmentation, likely due to massive amounts of technical debt. For such massive companies with over a century of operation, the task of modernizing IT infrastructure is daunting, but the cost of not doing so can also be massive.

WannaCry (2017)

What was the WannaCry hack?

Another incident that caught the public eye in 2017 was the WannaCry ransomware cyberattack, perpetrated by the Lazarus Group. In a matter of hours, WannaCry gated access to hundreds of thousands of machines running Windows in more than 150 countries. Attackers ransomed access to encrypted files in return for Bitcoin, often to no avail.

The attack largely affected hospitals in the UK, grinding all operations to a halt, as well as railway networks, and private companies. What is slightly more surprising than the rate at which the ransomware spread across the world is how it had made its way onto computers.

How did it occur?

A rogue group of hackers known as The Shadow Brokers leaked a number of tools and exploits that had allegedly been developed by the American National Security Agency (NSA). One of these exploits was EternalBlue, which took advantage of a vulnerability (CVE-2017-0144) in the way the Windows OS handles network packets. By crafting a malicious packet, EternalBlue allows remote attackers to execute arbitrary code on the machines. The Lazarus Group used EternalBlue to deliver another NSA-created tool, DoublePulsar that created a persistent backdoor on the breached machines. Using the backdoor, the group was able to install WannaCry.

After the theft of the exploits, the NSA informed Microsoft that they had known about CVE-2017-0144, which was quickly fixed. But as with Equifax, many organizations had not updated to the patched version, leaving them susceptible to EternalBlue. Fortunately for the world, WannaCry was killed in short order. Using a malware tracking system, one engineer noticed a domain name that he quickly registered. This domain was the killswitch for WannaCry and the reduced the spread of the virus to a mere trickle, allowing for time to deploy defensive measures.

What happened afterwards?

The exploits used by WannaCry were not relegated to just that incident. EternalBlue and DoublePulsar have been found in another major ransomware incident in 2017. The scope of impact these viruses have had prompted serious criticisms of the US government and the NSA, which had already been under public scrutiny after Edward Snowden blew the whistle on their operations. After the hack, the US Congress passed the PATCH Act, which balanced vulnerability disclosure and national security.

Cambridge Analytica (2018)

What was Cambridge Analytica?

Adding Cambridge Analytica to this list is cheating. It's not a hack as much as a data harvesting scandal, but it had deep implications for the cybersecurity community, and worth covering. In 2018, a whistleblower came forward about his tenure at Cambridge Analytica, a British political consulting firm, revealing a massive data harvesting campaign that straddled ethical and legal lines.

Much of the data gathered by the company was sold to US political campaigns that were able to create high definition psychographic profiles of the voting populace, which was subsequently used to deliver targeted ads. Later, investigations revealed that roughly 87 million Facebook users were affected.

How did it occur?

Nearly 300,000 users had their data scraped when they opted to take a personality quiz, downloading the app, *thisisyourdigitallife*. In doing so, users had agreed to abusive terms of service that allowed Cambridge Analytica to harvest not only their data, but that of their Facebook friends. At the very least, this data included the user's public profile, pages like, birthday, and location. In some cases, the extraction went deeper, allowing access to photos, timeline, and messages.

What happened afterwards?

The Cambridge Analytica scandal was exactly that, a scandal. By infosec standards, it was not a hack, as no passwords were stolen and all the data gathered happened with the consent of the user. Arguably, this outcome is even worse than a data breach. Being able to amass data at this scale without a cyberattack means the paltry protections around user data is a feature, not a bug.

Naturally, Facebook suffered consequences afterwards, including a $5 billion fine and various regulations. But on a broader scale, the scandal has forced the inevitable conversation about consumer data privacy. Look no further than the most expensive consumer protection legislation in the United States. The California Consumer Privacy Act, passed in 2018 cites that it is a direct response to Cambridge Analytica. Looking deeper, consumer facing companies have changed the privacy policies for third parties, minimizing their access to APIs or outright banning the use of cookies.

Capital One (2019)

What was the Capital One hack?

The Capital One hack shook security professionals given who the attacker was. Like most companies, Capital One hosts databases on Amazon, meaning those who have the most intimate knowledge of the cloud service are Amazon employees, not Capital One. This depth of expertise is what allowed an ex-Amazon employee to exploit a misconfigured web app firewall (WAF) to leak a heap of information such as:

- Over 100,000 social security numbers of US citizens
- 1,000,000 of Canada's equivalent to the US's social security numbers

- Tens of millions of credit card applications
- 80,000 bank account numbers
- Additional sensitive financial information

The hack was discovered quickly as the perpetrator publicized her actions, admitting to the leak on Github and Slack. By then, the damage was already done.

How did it occur?

Though details of the hack are held close to Capital One's chest, security professionals have come to a consensus that the vulnerability exploited was a server-side request forgery (SSRF). This class of exploit uses a server making requests while being under the hacker's control, in this case the compromised server was the misconfigured firewall. Supplying her own input, she requested credentials from Amazon's metadata services, which is architected to trust HTTP requests from known sources. In doing so, the attacker obtained AWS IAM credentials and accessed the Amazon S3 bucket that held Capital One's customer information.

What happened afterwards?

This hack brought much-needed attention to the SSRF vulnerability, which public clouds are particularly exposed to. Because cloud resources rely on varying degrees of trust and communicate mostly through HTTP, SSRF attacks can wreak havoc. Fortunately, invoking an SSRF attack requires in-depth knowledge of how cloud providers and their various systems work. Unfortunately, that does not make companies immune to it.

These types of attacks are not new by any means, but have grown in popularity among hackers of all colors due to the massive growth in cloud-host infrastructure and the usage of APIs in SaaS (and non-SaaS) companies. The Capital One hack has put pressure on cloud providers to introduce new measures in detecting SSRF vulnerabilities, but also adds further evidence in the case for granular privileges. After all, why should a WAF be able to request credentials that allow it to access anything beyond metadata?

SolarWinds (2020)

What was the SolarWinds hack?

Rounding out this list of prominents hacks is the most consequential hack of all time. Right on the heels of the winter holidays, FireEye published a blog post detailing their investigation into a supply chain attack delivered through Orion, an IT management and monitoring tool developed by SolarWinds.

It is difficult to grasp the scope of this hack. SolarWinds estimates that 18,000 of their customers could be exposed to the malware, ranging from nearly all Fortune 500 companies to the highest levels of government. Given that the intrusion remained undetected for six to nine months, it's safe to assume that attackers have gained persistent access to hundreds of internal networks.

How did it occur?

Perpetrators conducted a supply chain attack, meaning that victims were attacked by compromising a software supplier and moving downstream. Denominated as SUNBURST, the malware hid inside a digitally signed component that used a backdoor to communicate with third-party servers via HTTP. Because this component contained a digital signature from SolarWinds, it was considered secure. In any direct attack, it would be. But because this component was compromised in an update that SolarWind provided its customers, the trojanized component was already signed.

Once successfully infiltrated, the malware is able to transfer and execute files, reboot, disable services, profile networks, and exfiltrate data to a server via the backdoor. Being a well organized and sophisticated operation, SUNBURST went undetected for months. Embedded in a core component of the Orion framework, network traffic is masked as part of the protocol, which is further reinforced by the trust afford to a signed component.

What happened afterwards?

The fallout of the SolarWinds hack will persist for years and it will be many months before the damage is fully excavated. Considering that the depth of breach is still unknown, merely the breadth of organizations affected will require a massive effort to clean and inspect. Business will be disrupted, government slowed further, and confidential plans reassessed. The United States stepped up its cybersecurity efforts after the 2016 election on a national scale.

Would Teleport have changed anything?

Written earlier was the sentence, *"Cybersecurity as an industry is reactive to external threats. Practices evolve on the backs of hacks that change the way professionals think about security."* When reflecting on the best practices that cybersecurity professionals advocate for today, there are some important parallels between them and these 10 hacks.

Network segmentation

Perhaps the most common theme across all these hacks is the lack of network segmentation. Nowadays, professionals implore that networks employ micro-segmentation between each database, server, device, and application. This is largely due to the fact that trust within networks is deteriorating. In legacy models, activity within a private network is typically trusted. The assumption came from a period in which clients and servers existed in the same space, like an office, where it was easy to distinguish between authorized and unauthorized entry using IP addresses and metadata. But modern infrastructure includes cloud, physical devices, networked apps, and microservices.

Networks were never meant to manage authentication and authorization with the level of interconnectivity. All of the hacks described have exploited this weakness, gaining entry through one of the multitude of endpoints and hopping from database to database within a largely open network.

Reliance on secrets

Secret management has become another feature of modern infrastructure. In theory, each user should have their own secret for each resource, be it a key, token, password, etc. They should be rotated, stored in a hardened location, automated, and encrypted. Adhering to these practices is impossible. With each element of infrastructure being packaged into its own component and scaled up and down as needed, teams would find themselves buried under a mountain of secrets. Instead, they opted to share static credentials, either hardcoding them or leaving them lying around on the client machine. Hacks like Mt. Gox, Stuxent, and the DNC hack were able to find these credentials and impersonate authorized users.

Role Based Access Controls (RBAC)

Related to the secrets is the issue of RBAC. Often, secrets do not distinguish between those who have higher level privileges and those who do not. Admin credentials exist, but even normal users have access to data and resources they should not. Capital One makes a fine example here when the hacker used a compromised firewall to request credentials to vital customer data.

Experts counsel RBAC in accordance with the Principle of Least Privilege. Doing so would require machines to know information about who their users are. But unfortunately, when secrets are shared and consist of only a string and some metadata, like SSH keys or bearer tokens, enforcement is slippery at best.

Teleport for secure access

Teleport was built to manage authentication and authorization for modern infrastructure. It was designed around best practices, making it more resilient to the types of threats that these 10 major hacks suffer from.

- Teleport makes no assumptions about network activity. Between each server, application, database, and Kubernetes cluster stands a proxy that verifies the identity and permissions of each user. In doing so, any implied trust between resources and network is done away with. When a user needs to run a SQL query on a customer database, they must first go through a login flow with their identity provider. Outsourcing the authentication process to something like Okta or Auth0 ensures that each user is identified before being given access to anything.

- Instead of traditional secrets, Teleport uses certificates as credentials.These certificates are generated for each log in, have a time-to-live, and are signed by a separate certificate authority. In short, certificates do not suffer from any of the problems that secrets do. They do not need to be shared because they automatically generate. They do not persist because of a built-in expiry period, and can be easily discarded. This way, even if a hacker is able to infiltrate a device, their window of opportunity for data exfiltration is short, so hacks like Equifax would not have persisted.
- Encoded within these certificates is identity information, pulled from an identity provider after authentication. This extra data, like name, role, and team informs Teleport what the user is allowed to access. For example, the role of Jr. Dev may read production data, but not write. That level of granularity is not afforded with secrets.

Teleport is not a panacea. It’s likely that Teleport would not have prevented the SolarWinds hack, where the malware was injected upstream. But for Operation Aurora, Stuxnet, the Panama Papers, DNC hack, Equifax, and Capital One, Teleport would have at least mitigated the damage caused.