black hat EUROPE 2021
november 10-11, 2021
BRIEFINGS

The Bad Guys Win: Analysis of 10,000 Magecart Vulnerabilities

Nethanel Gelernter, Cyberpion

Cyberpion™

About me: Nethanel Gelernter

CEO/CTO @ Cyberpion

- Protecting the hyperconnected attack surface

Security Researcher / Hacker

- Web application & Cloud security
- Ph.D., hacks, research papers, talks, etc.

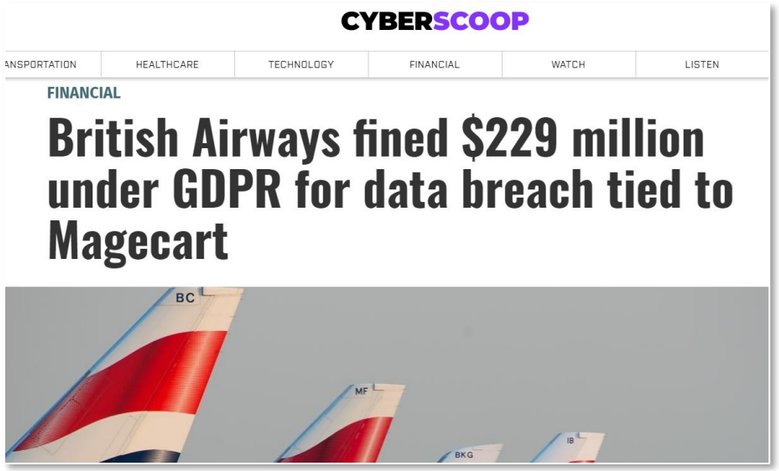
Agenda

- Background
- The Magecart problem and some statistics
- Protection mechanisms and bypasses
  - Why and how the bad guys win
- Summary
- Check whether your organization is in the statistics

https://www.cyberpion.com/blackhat2021

Background - Magecart

- Steal information by running malicious script in web applications
- Named after a famous hacking group
- Some famous attacks triggered anti-magecart solutions


CYBERSCOOP

ANSPORTATION | HEALTHCARE | TECHNOLOGY | FINANCIAL | WATCH | LISTEN

FINANCIAL

British Airways fined $229 million under GDPR for data breach tied to Magecart


SECURITYWEEK

INTERNET AND ENTERPRISE SECURITY NEWS, INSIGHTS & ANALYSIS

Subscribe | 2020 CISO

Malware & Threats | Cybercrime | Mobile & Wireless | Risk & Compliance | Security Architecture | Security

Cloud Security | Identity & Access | Data Protection | Network Security | Application Security

Home › Virus & Threats

Magecart Attack on eCommerce Platform Hits Thousands of Online Shops

By Ionut Arghire on October 11, 2019

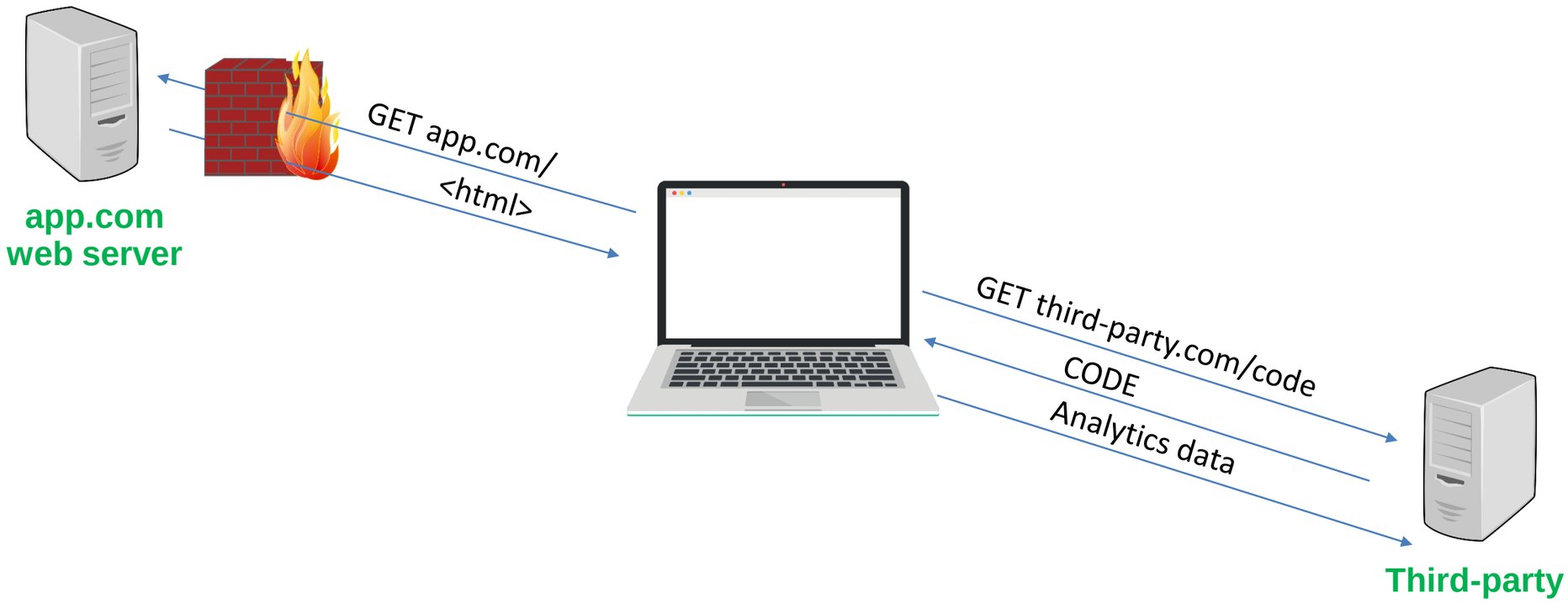
Share | Tweet | Recommend 0 | RSS

Magecart hackers have been gathering sensitive information from thousands of online shops after compromising top ecommerce platform and service provider Volusion.

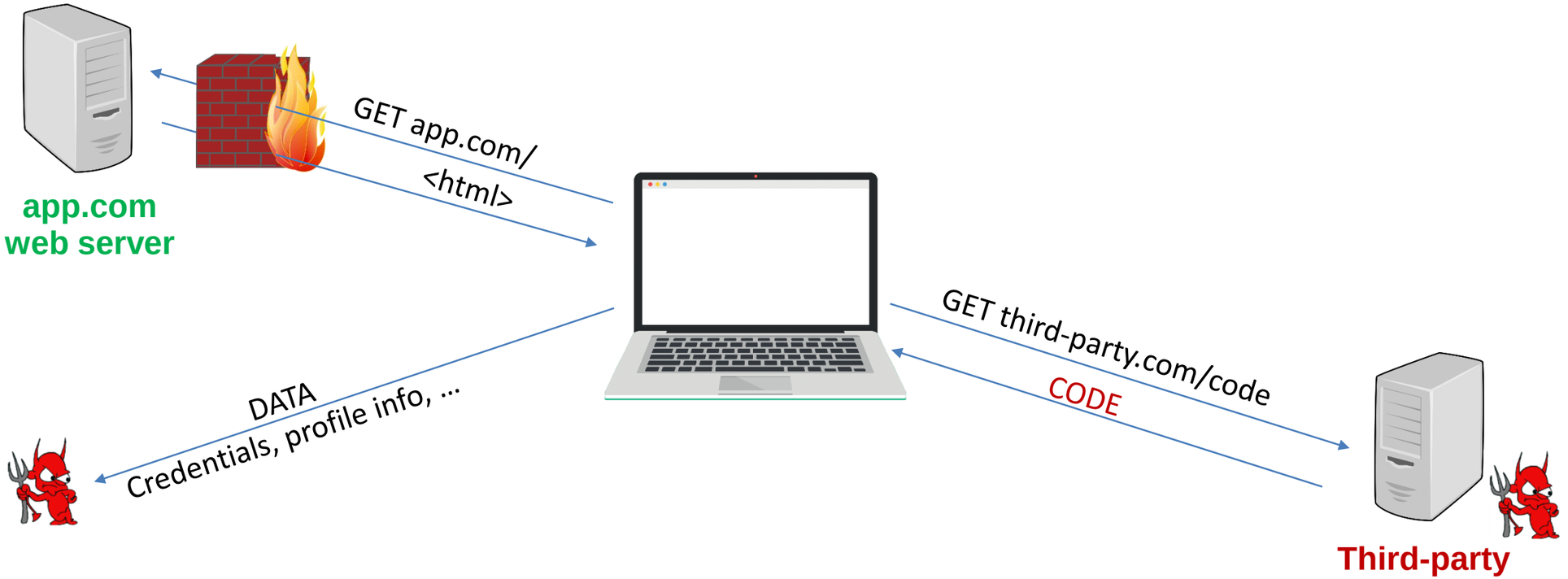
Background - Magecart in this talk

- Attack 3rd party from which script is loaded
  - Potentially, by many different applications
- Serve malicious script from the compromised 3rd party
- Steal information, but potentially do more than that

Background - Script Inclusion diagram



Background - Magecart diagram



Background - Magecart demo

- Example of vulnerable page

Magecart: Digital supply-chain attack

- Mistake of one vendor allows reaching many organizations
  - Example 1: misconfiguration in *marketing performance platform*
    - Strong indications that the misconfiguration was abused
    - We reported to one of the largest retailers in the US
    - Vendor fixed the issue right after
    - Many other retailers were affected but they don't know about it

Magecart: Digital supply-chain attack

- Mistake of one vendor allows reaching many organizations
  - Example 2: misconfiguration in one of the 5 largest advertising agency networks (according to Wikipedia) allowed to run code on many of their customers
    - Most of the largest brands in some countries, global corporations and many others
    - The issue has been resolved quickly after our report
    - Yet, as far as we know, the company has never informed its many affected customers

Background - Magecart challenges

Happens without the many affected organizations knowing about it

- Outside the organization's control
- 3rd parties do not disclose vulnerabilities/incidents
- Looks the same from the server's perspective
  - Attack could be completed without a single packet going through the organization's firewall/network
- Security solutions are blind about it

Statistics - The data

- Cyberpion's Global Monitoring System
  - Analyzing assets and the connections between them
- Detect Magecart vulnerabilities and track their exploitation
- In some cases, take active steps to hijack the hijacked assets and collect data from the 3rd party itself (and prevent the abuse)
- The trigger for the research: detection of several Magecart vulnerabilities/incidents that were missed by anti-Magecart solutions

30,000+ vulnerable assets

We made efforts to disclose vulnerabilities. Yet…

- We only had partial success
- In most of the cases, 3rd parties have not reported to affected organizations

→ Organizations do not know that they are/were vulnerable to Magecart vulnerabilities

- 15,000+ are still vulnerable/abused, and new issues are still being found

Popularity of affected sites

Vulnerable/abused site as of today:

- 15,000+ unique FQDNs
- 10,000+ unique SLDs
- Very popular sites (Alexa rank)
  - E.g., main pages of newspapers around the world, tech and retail sites
  - 1000+ under Alexa top 10,000 (including top 100 ones)
  - 2000+ under Alexa top 1,000,000

Cross-sector

- Some of the largest companies (top 5) and others in many sectors are affected:
  - Banking, Insurance, Healthcare, Retail, Pharma, Tech, Media/News, Energy, Industrials, Airline, accounting (“Big 4”), Food/Beverages, Telecom, Manufacturing, Electronics, Security, …
- Governments
- Universities
- Global organizations
- Some big companies are themselves the abused 3rd party
  - Their infra is vulnerable/being abused and expose their customers

Steal information

- Credit card numbers
- Login and personal information
- CRM information
  - E.g., from customized Salesforce login and other pages
- Customer tickets
- Web forms

Active abuse

- Change headlines in most popular media/news sites (stock exchange manipulation)
- Change advertisement/steal revenues from online ads services
- Download/install malicious files
- Act on authenticated applications

Threat model – Careful hacker

Professional hackers who are interested in money

- No defacement
- Hackers want to maximize benefits (info = money)
- Hackers want to stay unknown and to keep the attack unknown
- Hackers understand the Web

→ Hackers are careful as much as they can

The enterprise challenges

- Many websites/applications
- Distributed management
  - Many teams
  - Some are operated by 3rd party
- Complex/distributed applications
- Variety of websites/applications

Anti-Magecart solutions

- Script monitoring: periodically load the script and check whether it has been manipulated
  - Due to time constraints, and as these solutions are mostly native, only in high level
- Client-side solutions: scripts that are being added to the pages and inspect the scripts and report/block unintended behavior
- Browser-native: CSP and SubResource Integrity (SRI)
- Combinations of the above

General notes about the careful hacker

Once took over 3rd party, the hacker will not act recklessly

- Detect anti-Magecart solutions
- Study them carefully (will be discussed)

Attacks that worth millions - worth investment of hundreds of thousands

General things that careful hackers do

Start in small: change the script gradually

- Add spaces, unmeaningful changes, changes that are felt but are not malicious, and only then attack
- Start on a few targets and add more

Be non-deterministic

Decrease detection chance by making noise

- Trigger FP alerts on security solutions
  - How would you react on a real alert that comes after 10 FP?

Bypass script monitoring

Cautious hacker will always operate a controlled machine/service as usual before acting

- Learn traffic pattern
- Identify monitoring tools
- Correlate with scans bought/free demo
- Follow the manual/sites/pricelists (we scan every X minutes)
- Identify sources of requests (Cloud, Country) and avoid responding to them with the modified script. Attack only real victims' requests

Bypass script monitoring

Analyzing server-side logs of “protected” sites shows that it is super-easy to distinguish between legit traffic and monitoring

- Cloud or subnet
- Frequency
- Some are super naïve and add unique user agent
  - Mozilla/5.0 (Windows NT 5.2; WOW64; rv:21.0) Gecko/20100101 Firefox/21.0/VENDOR_NAME

Client-side solutions

General strategy:

- Monitor sources from which scripts are loaded and track their **behavior** – by running another script on the page
  - Static/dynamic analysis
  - Blacklisting/whitelisting

Client-side solutions

By the threat model – cannot work against the **careful hacker**

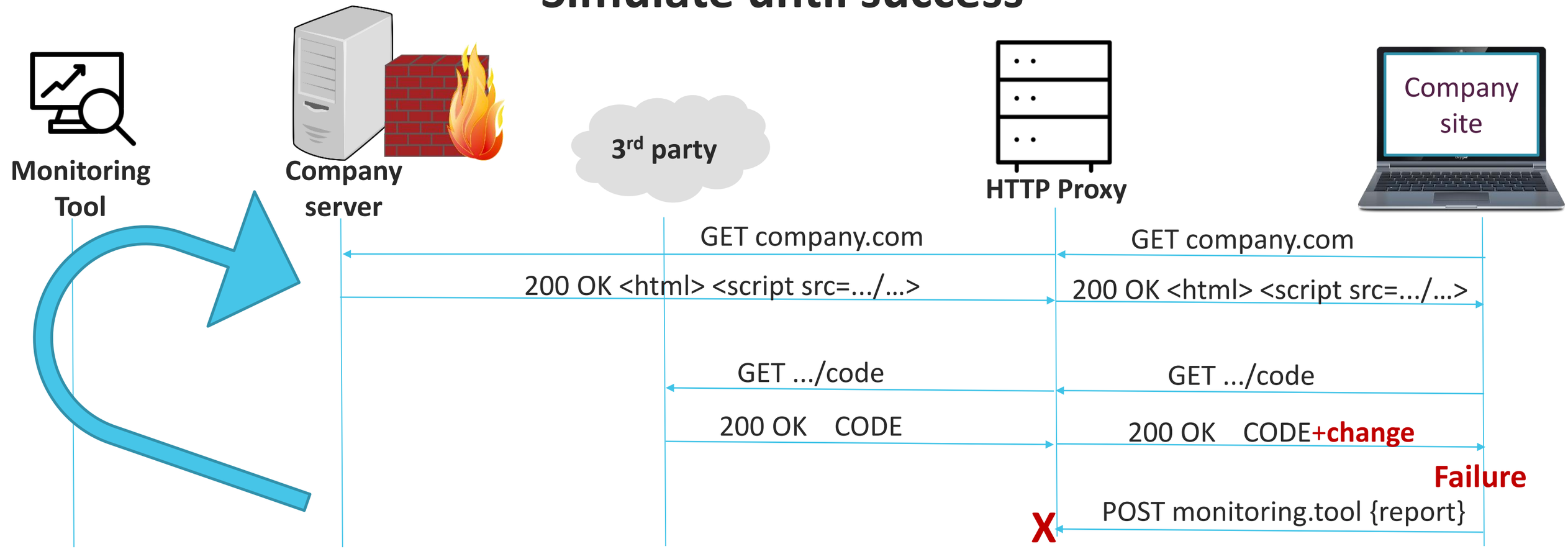
- Client-side solutions could be carefully analyzed
- Script of the solution vs. script of the hacker
  - Script of the solution is fixed
  - Hacker can simulate infinite number of versions and launch the attack only if his script wins
- Many frequently changing environments
  - Desktop and mobile browsers
  - Win in some of them is good enough
  - Different pages → different cases
- **Hackers can find the bypass and choose the vulnerable environment without being detected**

Bypass client-side solutions

Real world

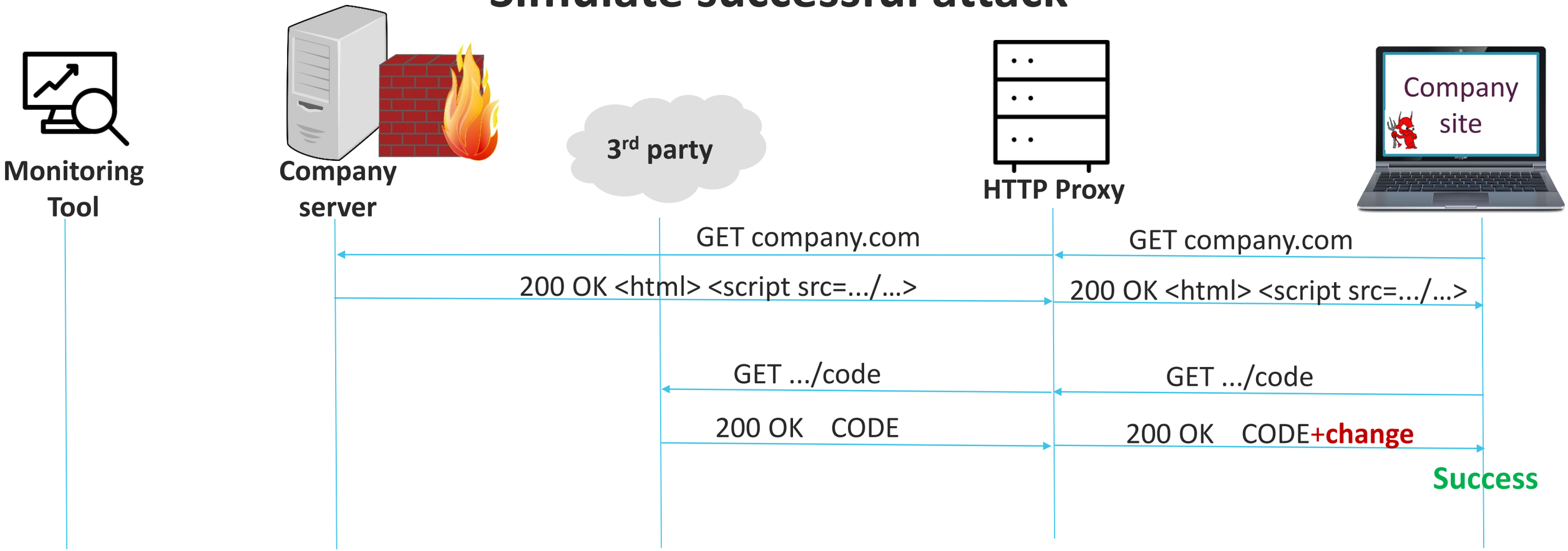
Bypass client-side solutions

Simulate until success



Bypass client-side solutions

Simulate successful attack

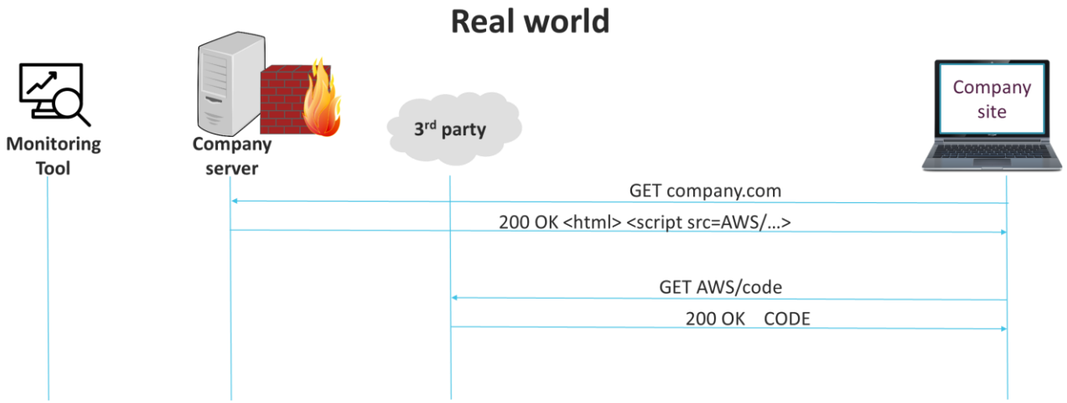


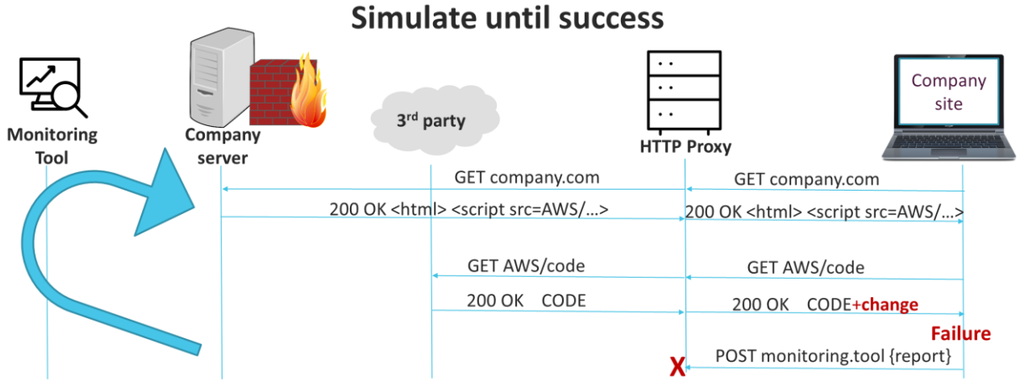
Bypass client-side solutions

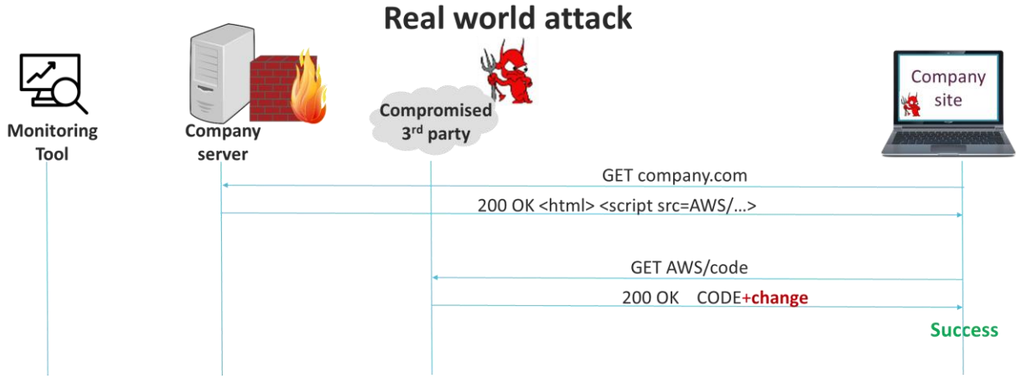
Real world attack

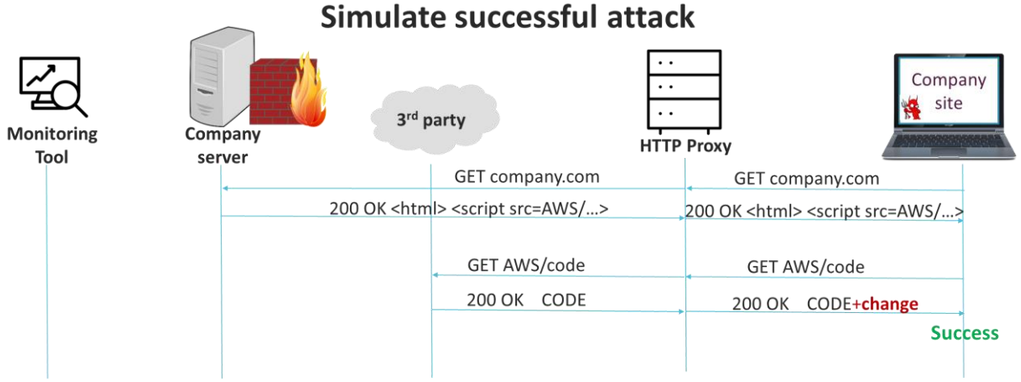
Bypass client-side solutions

Indistinguishable from the perspective of the monitoring tool and the company









Client-side solutions

Other disadvantages

- Performance effect
- Deployment required
  - Scale issue
- Solve the script problem by adding script

  → Can create other security issues, including of the same type trying to prevent
  - Remember abuse of anti-XSS mechanisms to launch XSS?

Client-side solutions

We found Magecart vulnerabilities in organizations that are protected that way

- We were able to use the methodology to reproduce attacks on “protected” organizations and document exfiltration of data
- We have not encountered a client-side solution that successfully stopped our Magecart attack effort
  - We succeeded in sending sensitive data, even to new third-party assets
  - Obviously, we can’t be 100% confident that the solutions did not detect the attack through a channel that we might missed

Use browser native mechanisms

- Content Security Policy (CSP)
  - The server instructs the browser to load script only from a whitelist of origins
    - Should be used correctly…
  - Limit the page to which the site can connect (e.g., send Ajax) to avoid leaking information
    - Could be easily bypassed for the use of leaking information

- SubResource Integrity (SRI)
  - Server tells the browser what is the hash of the scripts
    - If scripts are changed, the browser will not execute them

Use browser native mechanisms

- CSP is almost irrelevant when it comes to compromised 3rd party
  - The 3rd party is whitelisted… but now controlled by the hacker
  - Hacker can send the data to itself (the whitelisted 3rd party)
    - Data exfiltration could be done without Ajax in many ways
- SRI could have been a challenge, but it is not widely adopted and almost impossible to use with dynamic scripts
  - Remember: legit scripts change all the time

Enterprise use of browser native mechanisms

- Manual
  - Done for a fraction of the organization’s apps/sites
  - Usually, could be bypassed
    - At least in some of the pages

Enterprise use of browser native mechanisms

- (semi) Automated tools
  - Find the current use of scripts by using the CSP *report-only* mode
  - Define a policy that will prevent the use of new sources
  - Move to *block* mode
  - Done for a larger fraction of the organization's apps/sites
  - Here as well, could be bypassed
    - At least in some of the pages

Challenges

- Cannot be deployed strictly in scale
  - Variety of apps/sites managed distributedly by different people in different environments
- In “protected” sites/apps, not **all** the pages use SRI/CSP
  - For example, a script might be protected with SRI in one page and be without SRI in the other

The Trusted-Source Injection (TSI) attack

- Exploits the use of CSP or client-based scripts as done by tools/security solutions
- Does not exploit the implementation/design of CSP
- The idea: manipulate the tools during the learning phase to whitelist attacker-controlled origins

The Trusted-Source Injection (TSI) attack

- Inspect the target website and detect the *learning phase* of the tool
  - Either CSP-report or new script of a known anti-Magecart vendor
- Manipulate the page in a proxy level and add the required scripts
  - Preferably from different IPs
  - Many times
- Continue in the manipulation phase until the *learning phase* is over
- In CSP based solutions, it will be easier to see the success. In others, it depends on the implementation of the solution

Scriptless Magecart attack

Don’t touch the script, but other sources of data that affect its behaviour

- Most common: 4th-party to which Ajax request is sent, and that the 3rd-party script is affected by
- Examples (simplified):
  - eval(xmlHttpRequest.responseText);
  - value = xmlHttpRequest.responseText; divElement.innerHTML = value;

Scriptless Magecart attack

Example: scriptless Magecart vulnerability

- Common 4th party was Ajaxed
- 15 Alexa Top 10K sites and others
- Mainly media/news and retail
- 3 hours statistic:
  - **~500K requests** that could be abused from **~222K unique IPs**
  - 163K from iPhone, 128K from Android
  - 1000 from critically vulnerable browsers

Summary table: Enterprise vs. Hackers

	Script / Site Monitoring	Client-side script	Use of CSP & SRI
Scale to all sites (deployment issues)	✓	✗	✗
Scale to all pages on the same server	✗	✓	✗
Catches attacks that do not require script modification	✗	✗	✗
Hackers can't generate False Positive	✓[1]	✗	✗
If works, catches targeted attack (e.g. only specific IPs are attacks)	✗	✓	✓
No performance overhead or security risk	✓	✗	✗[2]
Enterprise wins the careful hacker	✗	✗	✗

Notes: 1) Not required. 2) Expected to break functionalities. Good only for not-changing static sites

Summary

Based on our analysis, we believe that

- Most Magecart attacks are completed without being detected
- Organizations later find their data sold or exploited.
- (script) Vendors do not report to affected companies

The bad guys win

What could be done?

- Use browser native mechanisms (CSP, SRI) carefully and manually on the most critical assets
- Browser vendors control the environment: new features can help!
- A different approach
  - Stay tuned
  - Follow myself and Cyberpion on LinkedIn
  - We hope to publish more information and more use-cases with more details once we let the many organizations check their status and fix the issues

Is my organization affected?

- Use your organizational email to check whether your organization was/is affected by one of the Magecart vulnerabilities

https://www.cyberpion.com/blackhat2021