



NTT Ltd. Global Threat Intelligence Center

Monthly Threat Report

November 2020

Contents

Feature article: Security in the app economy	03
Spotlight article: The Trickbot takedown	07
Spotlight article: Snapshot of threats to retail	08
About NTT Ltd.'s Global Threat Intelligence Center	09



Security in the app economy

Lead Analyst: Zach Jones, Sr. Director of Detection Research,
WhiteHat Security, US

It used to be simple; a retailer was a retailer and a bank was a bank. Initially, the role of software in non-technology sectors stayed behind the scenes, supporting the core competencies of that industry, like inventory management for retailers and account management for banks.

This is no longer the case. The trend of online consumer behavior which started in the dot com era and accelerated after the launch of the smartphone has forced business to compete to deliver their customer experience in a digital wild west. Competition has forced industries to take on a new and often unfamiliar role, that of a software shop.

In every non-technology sector, we hear the same and somewhat contradictory sentiment, 'We are an X company, not a software company, but our most important differentiation is the quality of the digital experience we deliver to our customers via applications.' We are all software companies now. Acknowledging this fact is critical to understanding why traditional IT security efforts have failed to control the risks introduced by organizations pushing the delivery of their digital business capabilities to new scale at increasing velocity.

Attack vectors and security spending are misaligned

According to our 2020 NTT Ltd. Global Threat Intelligence Report, 33% of observed attacks globally were application-specific and 22% of attacks were web-application based. This means a total of 55% of attacks detected globally occurred at the application layer.

According to Gartner Group, the 2020 Security Market Segment spend is about USD 59 billion annually. About USD 3.3 billion of that is associated with application security – or about 5.5%. Data from the 2020 NTT Ltd. Global Threat Intelligence Report suggests the threat to application security is somewhere around 55% of all attacks. Admittedly, this is not a complete risk evaluation, but if it is even close it suggests that application security spending should be more on the order of over 50% of security budget, instead of 5.5%. If the total security market spend remains unchanged, a 50% allocation to application security would mean about USD 29 billion, which is an increase of over 800% in spending related to application security. In even the shallowest analysis, this suggests that initiatives associated with application security are woefully underfunded.

Application security risks just don't look like traditional IT security risks

HTTP is the path of least resistance for developers to expose critical business capability. This is especially important

when organizations are trying to enable customer access in our 'there's an app for that' world. The problem is that represents a pipeline where benign and malicious traffic alike enter networks straight through firewalls and DMZs. The protocol was never designed for secure application delivery so building HTTP applications is prone to error. Threat actors will continue to abuse these virtual front doors and windows. They are easy to access and are often the weakest link in the security chain.

IT security is familiar with approaches which focus on controlling known risks against known components, such as:

1. Firewalls ensuring internal resources cannot be accessed externally.
2. Port scanning for services known to expose undesired access or capability.
3. Detecting systems with vulnerable unpatched software.

Out of the OWASP top ten application security flaws, **seven fall into the 'build' category of application risks.**

Application security does not fall in the same model as organizational controls. Application security risks can be simplified into three categories, the 'ABCs':

1. **Assemble:** Risk inherited whenever we bring together the components we rely on as the bedrock of our applications, like OS packages, frameworks and libraries.
2. **Build:** Risk created when we implement features without security by design or appropriate security controls.
3. **Configure:** Risk created when we deploy our applications to enable new functionality without hardening defaults and evolving past development setups.

Traditional IT security capabilities provide some visibility to risks in the assemble and configure categories, but almost no visibility into the risks of the build category. These risks are created by the features developed to meet specific needs of the business and the functions the application performs on behalf of a user. Notably, seven of the OWASP top ten application security risks are flaws which fall into the 'build' category of application risks.

Bolt-on security controls like IDS, IPS and WAFs can be effective at helping to manage well-defined IT security risks. Unfortunately, they often have inadequate out-of-the-box capability to understand the requirements of potentially complex web applications and potential risks they expose. This lack of context is especially true for IDS and IPS. WAFs can provide meaningful capability but require significant time and expertise in configuration, maintenance and monitoring. Securing a vulnerable application with bolt on techniques alone increases operational costs and leaves some risks unmitigated.

Application security testing

Unlike functional tests built specifically for the application they support, application security tests typically take the form of general tests that expose risks which fall into some or all of the 'ABC' categories. Commonly, application security testing is conducted by in-house security staff, through software-as-a-service vendors, or security service providers. Regardless of the method of delivery, a trifecta of techniques has emerged; DAST, SAST and SCA. Understanding the benefits and challenges of each technique will help you maximize your return on investment.

Dynamic Application Security Testing (DAST)

DAST tests a running application from the perspective of an attacker. The tools and techniques are similar to those used by threat actors. The most common DAST tool is a vulnerability scanner which crawls the user interface in attempts to discover the functionality of the backend server. The tool manipulates requests to the server to include simulated attacks. The goal is to cause the application to exhibit behavior which demonstrates evidence the application is vulnerable to common categories of application security flaws.

Automated DAST is best combined with manual testing to detect vulnerabilities across the breadth of the application. Automation enables the manual tester to focus on more complex functionality, including flaws which may exist within the business logic and security controls of an application.

DAST provides a view into the exploitable risks which are discoverable by an external threat actor who does not have inside knowledge of your application. It is an important baseline of your immediate exposures and informs necessary actions to reduce your risk profile. Automated DAST scanning will discover a different vulnerability profile than forty hours of manual assessment. Always consider the results in the context of the threat model which corresponds to the level of resources applied to your DAST evaluations. Threat actors may be willing

to devote far more than forty hours to attacking a high value application.

Benefits:

1. Vulnerabilities detected very likely to be exploitable risks worth paying attention to.
2. Continuously scanning production applications provides 'always-on' detection for newly introduced flaws and evolving threats.
3. Can confirm (or deny) the effectiveness of add-on security controls, application monitoring and vulnerability remediation efforts.
4. Mostly agnostic to your application's technology stack.

Challenges

1. Applications must be running in environments which tools and testers can reliably access. User accounts are required for best results. Coordinating environments and access can be difficult at the scale of a large organization.
2. Testing in production engenders an overly cautious approach which attackers do not share, leaving a potential gap between the vulnerabilities detectable by the good guys vs bad guys.
3. It is often difficult to achieve 'complete' coverage for all of a large application's complex functionality within a short release cycle. Since the testing is done without internal knowledge, it is difficult for the tester to know for sure that 'everything' has been tested.

Static Application Security Testing (SAST)

SAST analyses the application's source code. Automated analysis can be divided into three types: pattern matching, semantic analysis and runtime simulation. Each can provide value, but it is important to know that all 'code scanners' are not created equal. Pattern matching is easy to implement and runs quickly but suffers from a tradeoff between false positives and false negatives. More sophisticated analysis will have a higher implementation bar and

will run more slowly. The payoff is more accurate detection and results which are more consumable than single line pattern matches.

Automated SAST detects implementation flaws, not design flaws. Vulnerabilities can arise from both. Automation is best combined with manual code review during the software development lifecycle (SDLC) to detect implementation flaws across the breadth of the codebase. This allows reviewers to focus on the design aspects of key features and associated security controls.

SAST provides a view into the security hygiene of your application's codebase. Some of the vulnerabilities detected will be directly exploitable by attackers, while others will reveal weaknesses which add risk to your application in other ways. Its results are an important baseline for the application's level of defense-in-depth. If you only pay attention to critical and high-risk issues discovered by SAST you are missing this key part of its value. Proper review of SAST results can help support development of better habits and practices, and help identify both positive and negative trends in the development process.

Benefits

1. It is often possible to scan an application's codebase early in the SDLC reducing remediation costs.
2. Results in the form of code literally speak a developer's language. Developers are more likely to understand the findings even if they are not security experts.
3. Results highlight the root cause of vulnerabilities; making remediation efforts less of a scavenger hunt.
4. Complete assessments can be conducted in all but the shortest release cycles because clear measurements of scan coverage are achievable.

Application security is not just about patching. Effective application testing can help identify vulnerabilities and trends or tendencies which may potentially lead to the introduction of future vulnerabilities.

Challenges

1. The quality of the results is dependent upon the level of support and customization for your application's technology stack.
2. No awareness of environmental controls or trust architecture can make confirming the exploitability of findings challenging.
3. Security staff are often not code experts, so they end up relying on developers to configure scanning and triage findings. Developer enablement is great, but conflicting priorities can reduce overall effectiveness.

Software Composition Analysis (SCA)

SCA analyses the technology stack used by the application to detect publicly known vulnerabilities, age risk and license risk. Vulnerabilities detected by SCA are remediated via a patch or upgrade path.

Secure first-party code is easily undermined by vulnerable dependencies. Exploitation of these vulnerabilities can be as simple as running a publicly available script. Many highly publicized breaches in the past few years were the result of vulnerable dependencies used by the victim's application – vulnerabilities inherited with the tools and systems used in the development process.

SCA can provide a 'bill of materials' for your applications which can be easily searched for potential exposures when new zero-day vulnerabilities are disclosed.

Benefits

1. SCA can be conducted at almost any phase of project development.
2. Remediation is often a small change that requires no new code to be written.
3. Low cost but frequently results in significant risk reduction.
4. Feels like traditional IT security, detect unpatched components and upgrade them.

Challenges

1. Detection only as good as the sources of data on known vulnerabilities.
2. Vulnerabilities are often found in transitive dependencies which can easily result in 'we don't use that' confusion, because developers lack expertise with their dependency management system.
3. Vulnerabilities detected in dependencies for which the upgrade path has a breaking change can require significant refactoring of the application. This is rare, but can be painful.

Conclusion

An organization's security program is composed of a complex system of reinforcing controls. Managing that complex system is not always a simple process, so the best way to help make it more manageable is to help prioritize on the most effective controls – where is the biggest bang for the buck?

Year after year, detailed threat analysis tells us that one of the single biggest threat vectors we face: the aspect of organizations which is most likely to be targeted by external hostile actors, is the exposed web environment. This is the part of our environment which makes up our customer portals, our interfaces with the outside world which help communicate our message and helps us support our customers in meaningful, effective ways.

If we can apply efficient ways of helping to improve the security of those web-exposed systems, it can reduce an organization's exploitable footprint – making them less susceptible to an attack – or at least less susceptible to a significant attack which can have a negative impact on both the organization and their customers.

Application security is not just about patching. Effective application testing can help identify vulnerabilities and trends or tendencies which may potentially lead to the introduction of future vulnerabilities. Proper application testing is an effective preventive and proactive technique which can help reduce an organization's overall threat profile.

For more information on how NTT's WhiteHat Security Platform can help improve your application security see: <https://www.whitehatsec.com/drive-the-future/>.



#Spotlight 1



The Trickbot **takedown**

Lead Analyst: Global Threat Intelligence Center, US

NTT Ltd. collaborated with the Microsoft Digital Crimes Unit (DCU) and other partners to investigate and track activity related to the Trickbot botnet. On 12 October 2020, the Microsoft DCU moved to disrupt Trickbot, including active disruption steps to cut off key Trickbot infrastructure. This is an ongoing effort to help ensure that actors using Trickbot will no longer have the ability to initiate new infections or maintain activate infections already dropped into organizational environments.

Trickbot is a modular trojan which has been around since 2016. It typically infects victims via malspam and targeted phishing campaigns, and has been delivered as a payload by the Emotet trojan. Trickbot proliferates in an organization through the Eternal Blue vulnerability in SMB. It uses redirection and injection attacks to steal credentials and other financial information, and serves as a dropper for other malware, including additional toolkits and other malware like Ryuk ransomware.

Trickbot is highly modular, so can be used to perform a variety of functions. It spreads rapidly through an infected environment, including the targeting of vulnerable computers, IoT devices and routers. It uses obfuscation techniques to

help disguise itself and its actions. Based on available data, it appears Trickbot has compromised millions of systems and accounts.

Trickbot is available as 'malware-as-a-service', meaning it isn't just used by a single threat actor group, but by a variety of groups. Since Trickbot is modular, the exact functionality it supports can vary greatly depending on the exact threat actor group.

But, Trickbot is controlled via Command and Control (C2) servers managed by the operators of the botnet. These C2 servers are comprised of a variety of systems around the globe, including servers, IoT devices and routers. The C2 servers are used to issue commands to the infected nodes and control the flow of exfiltrated data.

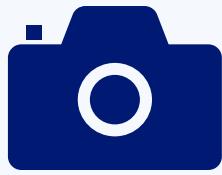
NTT Ltd.'s Global Threat Intelligence Center (GTIC) assisted the Trickbot disruption activity with months of analysis and research, helping identify, isolate and verify the C2 servers used to manage Trickbot operations around the globe. The Trickbot infrastructure was, and continues to be, complex. Researchers involved with the disruption analysed the techniques used in malspam and phishing campaigns, ongoing infections,

detectable C2 communications and other characteristics to identify as many of the Trickbot C2 nodes as possible.

On 12 October 2020, the initiative led by the Microsoft DCU took action with service providers and law enforcement on a global scale. The initiative had an immediate and measurable impact on Trickbot operations. Since the initial disruption, Trickbot operators have been focusing on restoring their normal operations and finding other ways to remain active. Since Trickbot operators are actively working to restore and replace lost functionality, the disruption activity continues to target the evolving infrastructure.

By 20 October 2020, the initiative had succeeded in eliminating about 94% of the global Trickbot infrastructure. It is likely Trickbot operators will continue efforts to restore functionality, just as disruption efforts to frustrate the botnet will continue. At least for now, the disruption activity has had a significant impact on the ability of Trickbot to threaten organizational environments.

Within eight days, the DCU initiative succeeded in eliminating about 94% of the global Trickbot infrastructure.



Snapshot of **threats to retail**

Lead Analyst: Jeremy Bender, Security Intelligence Writer,
Global Threat Intelligence Center, US


The retail industry has historically faced a varied threat environment due to its reliance on external connectivity, customer portals and the basic requirement of having a reliable and reachable web presence. Over several years, NTT Ltd. observed attackers most frequently targeting the retail industry with web application, application-specific, denial-of-service (DoS) and distributed denial-of-service (DDoS), as well as brute force attacks.


Typically, attackers launch DoS and DDoS attacks against retailers as either a way to gain notoriety or as a means of extortion; any amount of downtime, especially around major sales events, could lead to a significant loss of revenue. Web app, application-specific and brute forcing attacks, on the other hand, are largely intended to provide access to customer data, such as card payment information.


Attackers have also targeted retailers' internal networks in malware campaigns. Over the past two years, NTT Ltd. found trojans and droppers, spyware and keyloggers, as well as viruses and worms to be the most common malware types affecting the retail industry. Of the malware variants observed, remote

access Trojans (RATs) have most commonly afflicted retail. However, retailers have also been targeted with ransomware.

In August and September 2020, the hostile malware activity with the greatest potential impact on retail came from WannaCry, Ursnif and Emotet.

 **WannaCry**, which first appeared in May 2017, is a wormable ransomware. As WannaCry is self-propagating, its presence could cause extensive damage and downtime across an organization if the underlying systems have not been patched.

 **Ursnif** is an information-stealing Trojan which has existed in some form in the wild since 2007. As its source code was leaked, several Ursnif variants exist which are spread through malicious spam campaigns. An Ursnif infection can lead to the theft of credentials, financial information, email user accounts and more.

 **Emotet** was originally a banking Trojan which now has a range of additional functionalities. Emotet largely acts as a first stage dropper, leading to further infection with RATs, banking Trojans or ransomware. Like Ursnif, it is delivered via malicious spam campaigns.

While the threat environment for retail is varied, organizations can help mitigate threats by:

- Ensuring all underlying systems are patched and hardened, including by disabling default accounts.
- Instituting an aggressive patch management process.
- Use application testing to help proactively identify and mitigate potential vulnerabilities.
- Implementing strong password policies and ensuring accounts use multi-factor authentication, especially on critical or exposed systems.
- Standing up firewalls, web-application firewalls and filtering network traffic.
- Using updated antivirus programs.
- Ensuring all employees receive regular security training, with emphasis on phishing attacks.

NTT Ltd.'s Global Threat Intelligence Center

The NTT Ltd. Global Threat Intelligence Center (GTIC) protects, informs and educates NTT Group clients through the following activities:

- threat research
- vulnerability research
- intelligence fusion and analytics
- communication to NTT Group clients

The GTIC goes above and beyond the traditional pure research organization, by taking their threat and vulnerability research and combining it with their detective technologies development to produce applied threat intelligence. The GTIC's mission is to protect clients by providing advanced threat research and security intelligence to enable NTT Ltd. to prevent, detect and respond to cyberthreats.

Leveraging intelligence capabilities and resources from around the world, NTT Ltd.'s threat research is focused on gaining understanding and insight into the various threat actors, exploit tools and malware – and the techniques, tactics and procedures (TTP) used by attackers.

Vulnerability research pre-emptively uncovers zero-day vulnerabilities that are likely to become the newest attack vector, while maintaining a deep understanding of published vulnerabilities.

With this knowledge, NTT Ltd.'s security monitoring services can more accurately identify malicious activity that is 'on-target' to NTT Group clients' infrastructure.

Intelligence fusion and analytics is where it all comes together. The GTIC continually monitors the global threat landscape for new and emerging threats using our global internet infrastructure, clouds and data centers along with third-party intelligence feeds; and works to understand, analyse, curate and enrich those threats using advanced analysis techniques and proprietary tools; and publishes and curates them using the Global Threat Intelligence Platform (GTIP) for the benefit of NTT Group clients.

Our **Global Threat Intelligence Center** goes beyond a traditional research-only approach by combining focused research with detective technologies. This results in **true applied threat intelligence** to protect our clients with effective tools and services which reduce security risks and threats.

Recent assets



2020 Global Threat Intelligence Report

The 2020 NTT Ltd. Global Threat Intelligence Report (GTIR) is the culmination of the data the Global Threat Intelligence Center gathered and analyzed throughout the year. We produce this report by collecting a broad set of global data (log, event, attack, incident and vulnerability) to identify key cybersecurity trends of which businesses need to be aware.

[Download report](#)

If you haven't already, [register to receive the Monthly Threat Reports](#) directly to your inbox each month.

