SafeBreach

# SIMULATING A HACKER

# Table of Contents

# What's the Problem?

CISOs and their security teams have spent considerable amounts of time and money implementing best-of-breed technologies. Yet, attackers have never been more successful, and data has never been more at risk. Why? The real reason – one that is certainly difficult to admit – is that defenses have become so extraordinarily complex that security teams struggle to sort out the important issues from the noise. Enterprises typically deploy between 50 and 75 different security products on average, making it extremely difficult to understand whether security controls can stand up to an attack. Often the first time security teams know that defenses have failed is after actual breach has occurred.

To break this cycle, security teams can no longer rely on "best effort" security. Rather, security needs to be validated continuously to ensure that controllers are working as expected, alerts are firing when needed, and teams are prepared to provide resilience and response when a real attack occurs.

This technical whitepaper provides an overview of breach and attack simulation, and includes answers to frequently asked questions about how simulations actually work to challenge security controls.

# Validating Security

## TRADITIONAL METHODS

Security has always been a part of system architecture: Early LANs were 100% segregated from outside traffic, every host offered at least password-protected accounts, and access was typically only granted to trusted employees or users.

However, as interconnectedness drove business innovation, risk increased exponentially. Security started to move away from "best effort" into something that needed to be validated, quantified, and communicated – at least to internal teams.

During the last 10 years, security validation has evolved slowly:

- **Penetration testing:** Whether due to regulation or just security conscious teams, pen testing has a good goal, but is too shallow and infrequent to truly prove security effectiveness.
- **Vulnerability scanning:** Much easier than pen testing, thanks to automation, scanning is a good practice, but provides an even more basic view of security posture, based on open ports, missing patches, and a lot of supposition.
- **Red teaming:** Companies lucky enough to have Red Teams can be sure that they have creative, talented "internal attackers" that can create new attacks, and use their in-depth knowledge of internal controls and policies to find where holes exist, hopefully before they reach production.

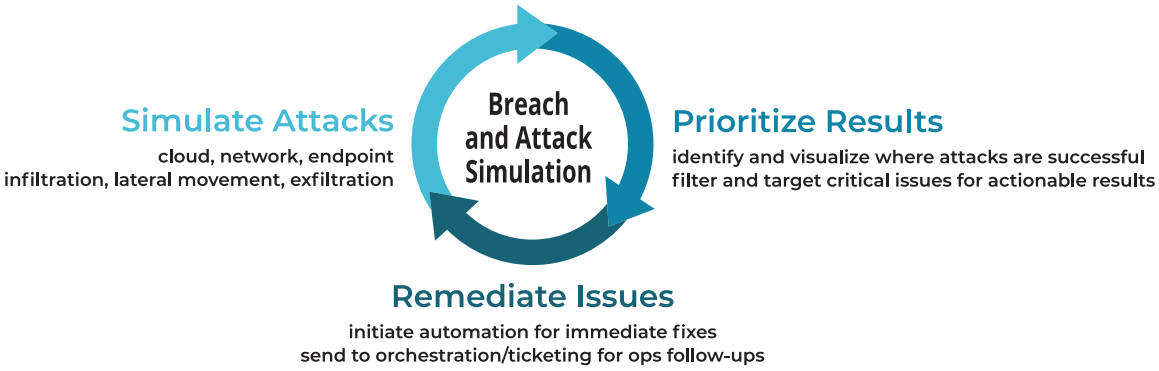
Each of these tools and techniques offer value, but as evidenced by the ever-increasing rate of breach, attackers still have the upper hand. These validation methods can't scale to cover the sprawl of today's modern production environments, are too often influenced by human biases, and are either too shallow, or take too much time to truly validate security across the entire kill chain.

## ENTER BREACH AND ATTACK SIMULATION

Security is a constantly moving target. Every day brings new risk, both external and internal. Externally, new attacks (and new attackers) are always emerging. Internally, security updates, patches, and configuration changes introduce the risk of human error.

Thanks to automation, Breach and Attack Simulation works continuously, and at incredible scale to simulate attackers and identify weaknesses in real time. This new approach enables data-driven security planning, minimizes exposure, and proactively identifies both where security is working, and where it needs to be bolstered.

## BREACH AND ATTACK SIMULATION OVERVIEW




Much more than just automating pen testing or red teaming, Breach and Attack Simulation should not only identify weaknesses, but also provide the insights, tools, and integrations to actually remediate findings.

- **Simulate attacks:** Unleash real attacks on production environments just like attackers do, but without harm or impact, to identify where defenses are working, and where they are failing.
- **Prioritize findings:** Quickly identify the right areas to focus on to stop the attacks most critical to your business.
- **Remediate security gaps:** Provide a seamless integration with operations teams or automation solutions to update configuration or otherwise block attacks, to incrementally improve overall security posture and effectiveness against threats.

# SafeBreach Architecture

The SafeBreach Platform is comprised of the following components:

- **Management server:** The centralized management server incorporates the complete Hacker's Playbook™ of breach methods, and manages a distributed network of simulators. Capabilities include the ability to manage all aspects of simulator configuration, review simulations that have been successful or blocked, and provide the ability to filter, prioritize, and analyze all findings. The management server can be deployed on-premises or in an enterprise cloud infrastructure.

- **Simulators:** The SafeBreach simulators perform the role of the attacker, simulating attacks across the cyber kill chain. Three different types of simulators are supported:
    - **Network simulators:** Network simulators are deployed as virtual machines within existing network segments. These simulators send real traffic, just as attackers do, to verify whether or not specific, granular breach methods will be effective against existing network security controllers and configuration.
    - **Endpoint simulators:** Endpoint simulators validate the effectiveness of endpoint security against various attacks and exploits. SafeBreach supports various Windows, Mac OS X and Linux operating systems and distributions with simple, lightweight agents for end user or server systems.
    - **Cloud simulators:** These are network simulators that act as infiltration and exfiltration points, located in the enterprise cloud infrastructure. Cloud simulators execute only network breach methods.

# Simulations Explained

## SIMULATING THE KILL CHAIN

The SafeBreach Breach and Attack Simulation Platform simulates hacker techniques to validate security. These simulations are, in actuality, real attack methods - made safe because they are only executed against SafeBreach simulators, and never use real production data. Instead, SafeBreach simulates data – such as credit cards, social security numbers, passwords, and much more. Simulations provide a complete kill chain perspective, and thus incorporate infiltration, lateral movement and exfiltration breach methods. A small subset of simulations in each phase is below:

### Infiltration phase

- Simulated malware drops
- Packed executables
- Malicious email

### Lateral-movement phase

- Simulating brute-force attacks
- Remote code execution
- Pass-the-hash

### Exfiltration phase

- Sending clear sample data over available ports
- Encrypting data to bypass security
- Trickling data within packet headers

## EXECUTING ATTACKS SAFELY

To validate network and cloud security, breach methods are executed between two simulators. Imagine a very simple example of a next-generation firewall segmenting two parts of an organization's environment – production and corporate. One simulator is placed in production, the other in corporate. SafeBreach will validate the effectiveness of that next-generation firewall by attempting to transfer, for example, a malicious payload from one simulator to the other. It's completely safe, but the NGFW should trigger appropriate threat prevention policies.

> ***Note:*** *Production data is never used. Instead, SafeBreach simulates the types of data relevant to the phase and type of attack used. Credit card data, customer record data, source code, hashed passwords and more are all simulated by SafeBreach, so customers can truly validate controller effectiveness without ever putting actual data at risk.*

Validating endpoint/host-based simulators includes network actions, as well as local methods such as dropping malware to disk, encrypting simulated local files, or executing remote commands. Again, simulations are safe, because malware isn't executed, or if performing an action like changing the registry, the actions are immediately reversed when simulations are complete. Endpoint security solution should stop these actions or trigger detection alerts.

## VALIDITY OF THE SIMULATIONS

A comprehensive set of breach methods spanning cloud, network and host-based methods are available. These methods are developed by SafeBreach Labs -- an elite team of offensive security researchers headed by Amit Klein, VP Security Research and Itzik Kotler, CTO and co-founder, SafeBreach. SafeBreach Labs incorporates expertise in red team security with forensics, threat research and national cyber security, and focuses on the following:

- **Analysis of attacks in the wild:** We research attacks in the wild and break them into individual breach methods. This process is automated for efficiency, allowing us to react very quickly to attacks in the headlines.
- **Active research:** In addition to existing attacks and breach methods, our team also proactively conducts research to identify new vulnerabilities or attacks. This active research is shared with the security community in conferences such as Hack in the Box, Black Hat, BSides etc.
- **Threat intelligence:** Enterprises that already have a subscription to threat intelligence feeds supported by SafeBreach can choose to transform the indicators of compromise (IoC) to breach methods.
- **MITRE ATT&CK collaboration:** The SafeBreach Labs works closely with MITRE on new attack techniques. Attacker techniques that have been identified within the MITRE ATT&CK framework are designated appropriately within the SafeBreach playbook for security teams that are aligned to this adversary model.

# Frequently Asked Questions

### *How safe are these simulations?*

SafeBreach was designed from the outset to run safely in production environments. To highlight the safety of simulations, we'll look at what occurs during a malware transfer simulation and malware payload drop.

During a malware transfer simulation, SafeBreach creates an artificial malware model using reverse engineering that mimics the behavior of a malware sample, for example, writing files, encrypting local files, opening a socket, and attempting to communicate externally. But, this reverse engineered "malware" will not do any damage. In the malware download and drop test, SafeBreach uses a real malware sample that is downloaded from a "server" simulator to a client simulator and actually saves it to disk. Once we have confirmed this action is blocked or allowed, the sample is then removed.

SafeBreach also simulates attacker exploits. For example, we support Meltdown simulations that read kernel memory, fileless Mimikatz injection using Powershell, WannaCry exploits (Eternal Blue), and remote exploitation of Apache Struts server vulnerabilities. These exploits are kept safe by sending malicious packets that the real exploit would have sent, but containing the impact to our own simulators, not actual in-production devices or applications it was meant to exploit. A security device such as an IPS or IDS will still recognize the exploitation packets as malicious, but no harm has come to the environment.

### *How do simulations trigger my endpoint security controls (detection and/or prevention)?*

SafeBreach validates both prevention and detections controls as follows:

- **Prevention:** Since we send real traffic between simulators, and run real processes on endpoints, all prevention controls should actually stop our simulations from occurring if deployed and configured correctly. When this occurs, SafeBreach will show that those specific techniques are blocked and which techniques managed to bypass prevention controls.
- **Detection:** Detection technologies, by design, don't stop attacks, but rather alert on malicious or suspicious actions. Our methods are purpose-built to also trigger these types of alerts, because our methods are indeed the very same actions real attackers use. Our methods include "compromised behavior" as you'd expect to see from a compromised endpoint. (For example, trying to communicate to command and control, trying to install further payloads, attempting to brute force passwords, or sending hashed credentials).

  Detection rules will fire when these methods are run, but since the methods were not stopped, SafeBreach will show they are successful. That said, since SafeBreach integrates with SIEM, it is easy to correlate a specific attack with the expected alerts, to validate that all allowed methods have a corresponding alert for SOC teams.

### *I have an ephemeral environment - how can this help?*

Having virtualized environments, with "disposable" machines or microservices can indeed help prevent attackers from establishing footholds in an environment, and can make targeting data more difficult for bad actors. In these types of environments, the assumption is that by having machines or services only exist for a short time, the attacker will not have enough time to exploit them.

While the containers or machines may be virtualized, a network must exist at some point for them to communicate. Physical machines, or lower level virtual machines, must indeed exist as well, in order to host these ephemeral entities.

Validating the security at the network, and host machine level will ensure that attackers cannot target the parts of the environment with longer lifespans. And of course, simulations can validate (or invalidate) the assumptions that are critical to this type of infrastructure – ensuring that the actual configuration and deployment lives up to the architectural ideals of security.

### *Who develops the SafeBreach Hacker's Playbook™ of methods?*

All SafeBreach methods are developed by SafeBreach Labs – an internal team of dedicated white-hat hackers.

New methods are created via external monitoring of the hacker underground, sourcing of intelligence feeds and collaboration with security research teams. Additionally, the team proactively identifies new and unique breach methods as part of our research and development effort to "simulate the hacker". This research is shared with the security community at conferences like Hack in the Box, Black Hat, RSA and DEFCON.

### *What if I have [home-grown security solution] that SafeBreach can't possibly know about?*

Many SafeBreach customers have implemented custom or semi-custom security controls. SafeBreach is a "black box" style tool, i.e. it should not "know" the state of the security it's attempting to validate. Just like an attacker, SafeBreach doesn't know the particular layout of a network, or the types of hosts within it. Instead, it simulates attacks to see what works - regardless of what controls are in place. If the controls are effective, then the methods will be stopped. If not, the methods will be successful.

### *What's a typical deployment?*

Simulating attacks in automatic, and initial deployment is simple and fast. Typical deployment, including initial simulations and findings, is completed in under an hour.

Deploying and configuring simulators is very simple, and requires only definition of the following:

- **Roles:** Simulators can play the role of infiltration, exfiltration or critical service points, and can be defined with these roles simply, from the management console. These can be changed with just a click, to affect the types of simulations that are performed. Infiltration nodes are where simulations are initiated, while exfiltration nodes are where simulators attempt to might extract data assets to. Critical services contain important data assets, and represent the targets that need protecting.
- **Types of breach methods:** By default, the SafeBreach platform runs all breach methods to validate security against both known and unknown attacks. Attacks can also be assembled into one or more custom "matrices" of attacks if necessary, to validate a specific security control, or specific scenario or specific attack type/phase. For example, using only executable file download simulations to check the efficacy of a secure web gateway product.
- **Communications channel between simulator and management server:** The SafeBreach deployment does not require any understanding of the topology or configuration of the enterprise environment. However, simulators need to have a communication path to the management server to ensure that simulations details can be shared with the management server. The management

server will reach out to cloud simulators, and internal network or endpoint simulators will reach out to the management server.

SafeBreach typically executes its complete database of breach methods (unless configured otherwise), and initial results can be seen in less than 20 minutes. Breach methods can also be rerun after remediation of a particular issue is complete.

***My red team is always coming up with new/novel attacks, how can you simulate the things I do?***

The SafeBreach platform includes the ability to add custom breach methods. Additionally, SafeBreach Labs is constantly updating the Hacker's Playbook with new and novel methods that they develop, and releasing new methods based on new attacks in the wild.

# Use Cases

Breach and Attack SImulation from SafeBreach helps our customers do much more than simply find security weaknesses. By simulating the hacker, prioritizing findings, and taking immediate action, organizations can:

### Get more from existing security

Security controls are incredibly flexible, but are often deployed with generic "one-size-fits-all" policy recommended by vendors, or configured once and never revisited. Breach and Attack Simulation safely simulates thousands of attacks to see which policies are effective, which need to be updated, and where holes exist. By optimizing config and ensuring controls work in concert, security teams can get the most from existing security investment.

### Minimize security exposure

Enterprise environments are far from static – constantly updated to meet the needs of the business, and to stop new and emerging attacks. However, all this configuration often leads to simple oversight, or human error, that can introduce risk. Thanks to continuous validation, Breach and Attack Simulation identifies new exposure in hours, so security teams can minimize exposure time and prove the effectiveness of new configuration.

### Prepare for audits

Annual penetration test and compliance audits bring stress and risk for CISOs and security teams. These tests often result in a list of findings that's too long for operations to address, and is only representative of a small window of time before changes to the environment make it obsolete. Breach and Attack Simulation runs continuously, to find risks well before audits, and smooth the process of maintaining compliance.

### Test alerting and action plans

Every security team knows that defenses are built from people, processes, and technology, but often the technology receives all the focus. By simulating attacks, SOC and MSSP teams can perform breach scenario training before a real attack occurs, to validate action and alerting plans.

### Rationalize security investment

Security investment is too often a “gut feel” based measure, and too often executive teams only start deep security investment after breach has occurred. Breach and Attack Simulation provides real security data, to justify further security investment, and to address the growing issue of proving security against headline attacks.

With Breach and Attack Simulation working continuously, security teams will have the data they need to improve and maintain security, without guesswork, or reliance on vendor claims.

# Learn More

To learn more about how automated, continuous breach simulation can help financial companies answer the question, “are we secure?” check out these additional resources:

- **Breach and Attack Simulation Primer**
- **Ten Things to Look for in Breach and Attack Simulation**
- **The SafeBreach Hacker's Playbook Findings report - 3rd Edition**

**ABOUT SAFEBREACH:**

SafeBreach is a pioneer in the emerging category of breach and attack simulation. The company’s groundbreaking platform provides a “hacker's view” of an enterprise’s security posture to proactively predict attacks, validate security controls and improve SOC analyst response. SafeBreach automatically executes thousands of breach methods from an extensive and growing Hacker’s Playbook™ of research and real-world investigative data. Headquartered in Sunnyvale, California, the company is funded by Sequoia Capital, Deutsche Telekom Capital, Hewlett Packard Pathfinder and investor Shlomo Kramer. For more information, visit **www.safebreach.com** or follow on **Twitter @SafeBreach**.