MENLO SECURITY

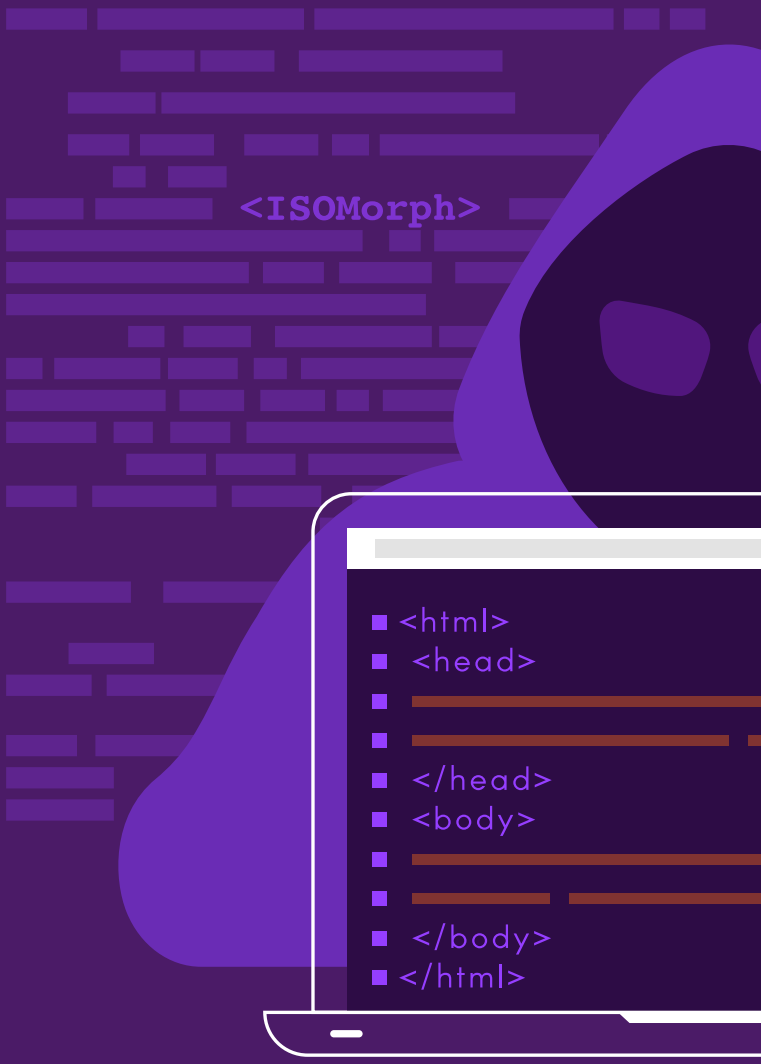
# Anatomy of highly evasive threats:

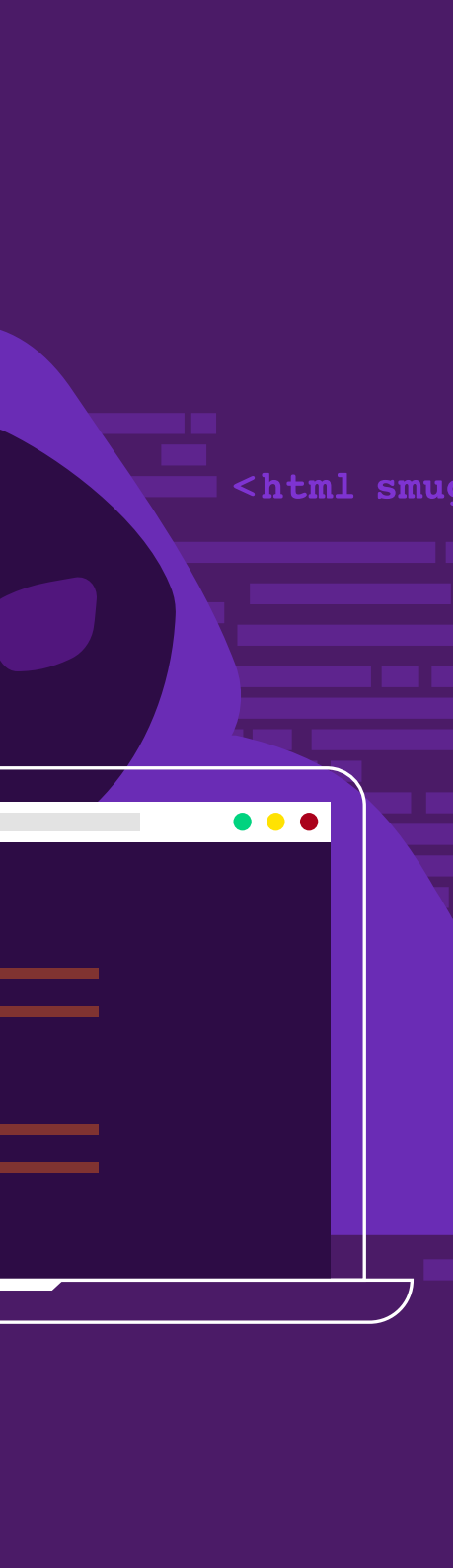
## 4 ways threat actors are getting past your security stack


eBook

# Page **Contents**








# The role of antiquated security tech **in the emergence of HEAT attacks**

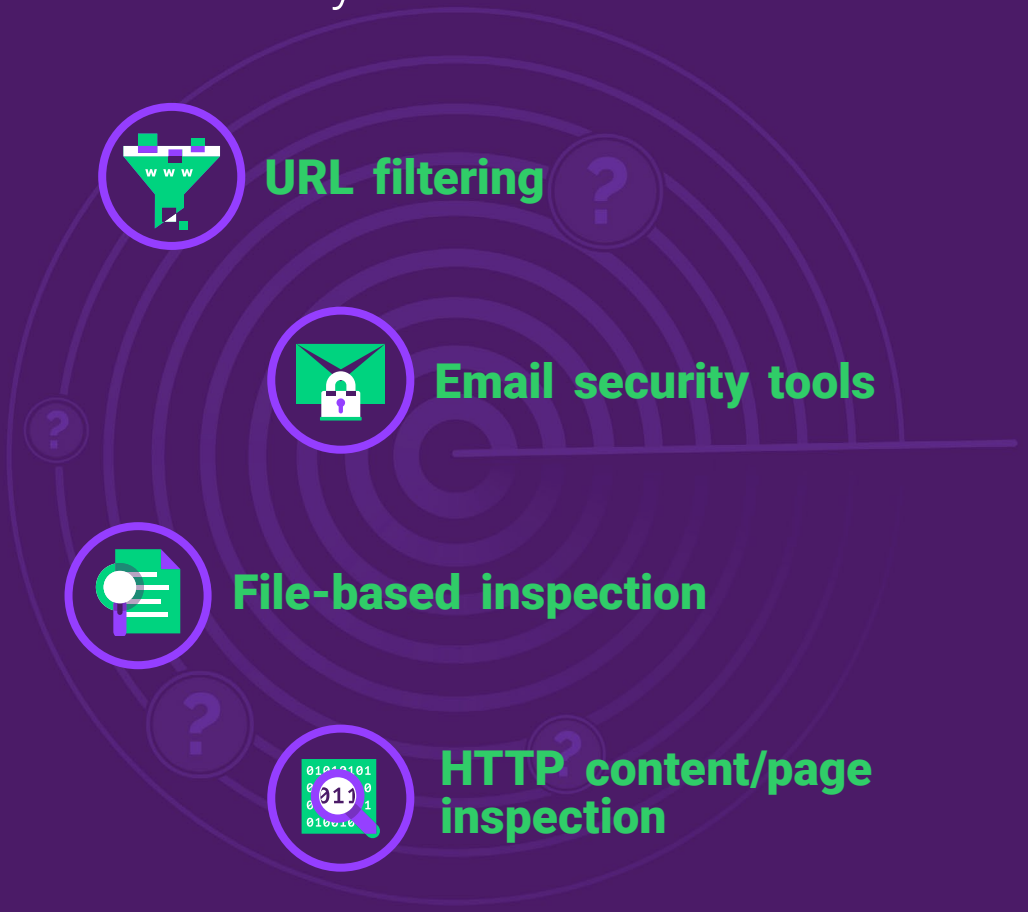
The way that we work has changed dramatically over the past few years, but security hasn't. Today, knowledge workers spend most of their time in web browsers accessing business-critical applications, communicating and collaborating, and ensuring the business moves at the speed its required to. Thanks to this productivity tool, they're increasingly doing work outside of the confines of an office. Yet current web security technology was built for a time when remote work was rare, and the web browser consisted of a simple interface for browsing websites. The detection-based network and endpoint security tools many organizations rely on, such as Secure Web Gateways (SWGs), firewalls and sandboxes, were initially intended to protect networks and endpoints, and they're now failing to meet the demands of modern work. Organizations that rely solely on detection-based security technology are leaving the door wide open for evasive cyber attacks that easily bypass these tools.

In order to detect threats like ransomware, legacy security tools need to analyze all traffic on the network and decide whether or not it's malicious. This creates costly and time-consuming false positives and negatives, as these tools aren't going to get it right every time given the sheer volume of traffic. Security has to be right every time in order to stop a breach, but threat actors only have to slip through the cracks once. By the time the threat is detected, it's already done its damage and the system is compromised.

Attackers know this all too well and have been focusing on evading detection with a class of threats that leverage the web browser as an attack vector to gain initial access to the network and steal credentials or deploy malware, often ending in a full-blown ransomware attack. Largely unknown in the cybersecurity industry and unprotected against, these **Highly Evasive Adaptive Threats (HEAT)** were built specifically to compromise the shortcomings of traditional security tools — and that's exactly what they've been doing. As security scrambles to catch up to HEAT attacks, it's becoming clear that attackers have the upper hand in the constantly evolving cat-and-mouse game that cybersecurity is.

In this ebook, we'll be diving into **how attackers leverage each of these HEAT characteristics to slip past traditional security tools** undetected and gain free reign over the network.

**HEAT attacks are able to evade the following traditional security features** and technology currently in use today.



# Attack method 1:
# **Evading URL filtering**

## **What** it is.

Letting workers navigate to any website they want to is a recipe for disaster considering how many new malicious domains threat actors are spinning up every day. Categorizing websites as trusted or untrusted based on domain age, reputation, and popularity, among other criteria, has been the traditional approach to solving this problem. Threat actors are all too familiar with this approach, and are **easily bypassing it** by creating or gaining control of websites categorized as trusted then leveraging them to deliver malicious content.

# **How** it works.

Threat actors are **evading URL filtering technology** through the following techniques:

## Compromising benign websites

One way threat actors evade URL filtering technology is by compromising poorly secured websites that have already been categorized as trusted by web categorization engines. Once the trusted site is taken over, **all it takes is the flip of a switch** to get it to start deploying malware or stealing users' login credentials. Attackers often remove the malicious content just as quickly as they put it up — making the attack even harder to detect.

## Hiding malicious content behind visual CAPTCHAs

Many legitimate websites use CAPTCHAs to protect themselves against API bots and ensure that only humans are visiting the site. However, this also blocks web categorization crawlers which are unable to fill out the CAPTCHA, **essentially putting up a wall that attackers can hide malicious content behind without it being inspected.** Users might think a site is secure when they see a CAPTCHA, but this isn't always the case.

## Building temporary malicious sites

Instead of taking over a trusted website, **threat actors can also register domains themselves** and build up their reputation until they're categorized as trusted. They may even build up repositories of trusted sites in their control for use in later malicious campaigns. Once the attack is complete, threat actors can even switch the site back to its original trusted state to use it in another attack.

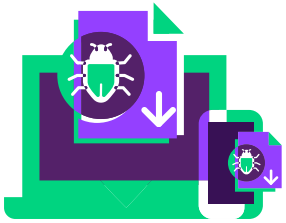
## Multi-factor authentication (MFA) bypass

In this attack, traditional security tools cannot detect the **interception of MFA credentials and tokens.** Prevention is challenging due to the attackers' speed and surprise. Even if a fake sign-in page is blocked, gangs can quickly create new domains to target others. Deny-listing, URL filtering, and phishing training are insufficient, and the MFA process, often using SMS, remains hidden from security teams.

# **Attack** in action.

**01.** Threat actor takes over a website categorized as trusted or builds one themselves.

**02.** Trusted site is rigged with malicious code. SEO is sometimes used to drive page ranks of compromised sites up in search engines, making it easy for users to find and navigate to these sites.



**03.** Victim visits a seemingly normal website and downloads malware or has their login info stolen.

## What technology it evades:

- Web categorization technolgy
- URL reputation engines
- Legacy SWGs

# Attack method 2:
# **Evading email security tools**

## **What** it is.

Workers can't fall victim to a phishing attack if they don't click on a malicious link, and they can't click on a malicious link if it's prevented from ever reaching their inbox. That's the logic behind the **malicious link analysis technology** leveraged by email security tools like Secure Email Gateways (SEGs). But that logic falls apart when threat actors have evolved their tactics to target workers with phishing attacks beyond email.

# **How** it works.

Threat actors are **evading email security tools** through the following techniques:

## Attacks delivered via collaboration platforms

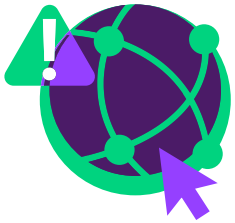
We expect phishing attacks to hit our inbox, so we're naturally suspicious of emails that seem out of place. **We're not expecting them through text messages, social media, and collaboration platforms like Slack** — we tend to trust those contacts more. Threat actors take advantage of this implicit trust by impersonating one of your contacts or compromising their account, and sending you malicious content that's already past your browser and out of the reach of malicious link analysis.

## Malicious Office docs and PDFs containing malicious links

SEGs only analyze links in the body of the email. They're not looking for links inside documents or PDFs, so they won't find them if that's where they're hidden. **Users trust Office documents and PDFs, and are more likely to click on them and inadvertently download malware or ransomware.**

# **Attack** in action.

**01.** Attacker impersonates trusted contact and sends malicious content through a channel outside of email.



**02.** User clicks on malicious link.

**03.** Malicious content is downloaded to the endpoint.

## What technology it evades:

- Secure Email Gateways (SEGs)
- Malicious email link analysis engines

# Attack method 3:
# **Evading file-based inspection**

## **What** it is.

By leveraging legitimate web browsers, threat actors are crafting attacks **designed to slip past legacy file-based inspection technology** by making the malicious content impossible to analyze. They know that legacy security technology can't completely ensure every file is safe, and they're taking advantage of the file exceptions made by inspection engines.

# How it works.

Threat actors are **bypassing file-based inspection engines** through the following techniques:

## Oversized files

A completely secure sandbox would require every file coming in to users to be inspected — a horrible prospect as far as productivity is concerned. As a work around, **sandboxes don't inspect** files over a certain size, some of which are rejected, and others which are sent straight to the user. That's where threats sneak in unnoticed, skating by this first layer of defense without being analyzed at all.

## Password protected archives

In a similar tactic, malicious content is hidden behind password protected docs that can't be inspected by traditional security technology like sandboxes. In order to protect sensitive information like payment card information (PCI) data and personally identifiable information (PII), it's exempt from sandbox inspection. Unfortunately, threat actors can easily abuse this security measure by hiding malicious content in documents that appear to contain sensitive information.

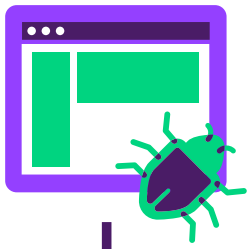
## HTML smuggling

Threat actors are leveraging legitimate web browser features like HTML5 and JavaScript to **hide malicious content directly in HTML code.** The content is hidden in a JavaScript BLOB (binary large object), which is then reassembled on the user's device dynamically within the web page. The malicious content is built on the endpoint once it's already past the firewall, anti-virus (AV) technology, or sandbox — completely evading detection by these legacy security technologies.

## Additional HEAT techniques that evade content inspection

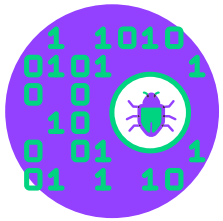
- Embedded scripts inside MSI installers, archives etc.
- Multiple redirections leading to a file download event

# **Attack** in action.



**01.** Malicious file is constructed at the browser without being inspected.

**02.** Malicious file bypasses legacy security technology like firewalls, sandboxes and anti-virus engines.



**03.** File types normally blocked by Secure Web Gateways (SWG) can still compromise endpoints without any interaction from the user.

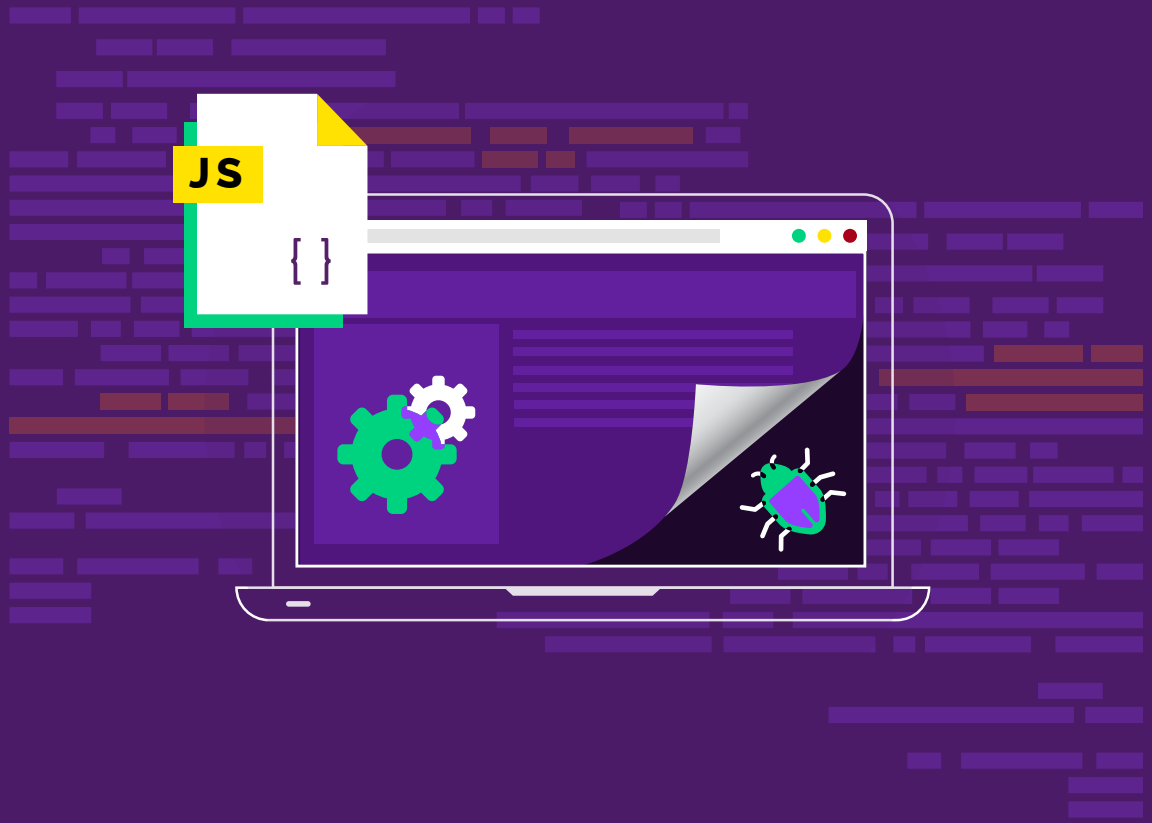
## What technology it evades:

- Anti-virus technology
- Sandboxes

# Attack method 4:
# Evading HTTP Content/Page Inspection

## **What** it is.

Regardless of how threat actors try to hide malicious content on websites, it ultimately has to exist somewhere in the source code of the page. HTTP content and page inspection engines scan the entirety of this source code for malicious content. But, of course, scanning for malicious content doesn't guarantee you caught everything. Threat actors have figured out how to **get around this security technology** by taking advantage of features that allow legitimate websites to obfuscate sensitive code.



# **How** it works.

Threat actors are getting around HTTP content and page inspection engines **through the following techniques:**

## Obfuscated malicious JavaScript

In a similar way to how HTML smuggling attacks work, threat actors hide malicious content in JavaScript code, which then reassembles itself on the endpoint after passing through HTTP traffic inspection. There's no malicious content to detect because it doesn't exist in a form that can do any damage until it's created on the end-user's browser.

## Dynamically generated phishing logos

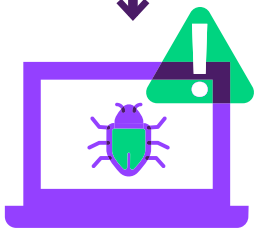
Attackers are manipulating web page source code to obfuscate malicious phishing logos. Often beginning with a sophisticated phishing page, these attacks use creative CCS trickery to hide known phishing logos under seemingly harmless images to avoid visual detection. The phishing logo appears on the end-user's device without being analyzed at all by HTTP traffic inspection engines.

## Additional HEAT techniques that evade content inspection

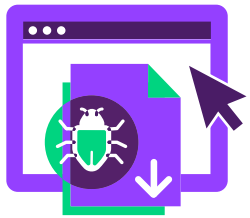
- Evading phishing kits by encoding or leveraging individual functions that make the extraction of page resources for analysis more difficult.

# **Attack** in action.

**01.** Attacker obfuscates malicious content via JavaScript or CCS.



**02.** Malicious content is reassembled on the endpoint.



**03.** Victim clicks on phishing logo or downloads malware.

## What technology it evades:

- HTTP content inspection technology
- Legacy SWGs

# Preventing **HEAT attacks**

Traditional security tools simply aren't built to stop HEAT attacks. It's no surprise that they're consistently failing in the face of these modern threats — even with increased cybersecurity budgets. Their reliance on detection narrows their view of threats to the ones they've detected on the network. This gives attackers a clear blueprint: evade initial detection and you're in. That's all that's needed to gain free reign over the network to move laterally, steal data or compromise critical systems.

It's become painfully clear that an entirely new security paradigm is needed. So what's the best way to stop HEAT attacks? Making them never happen in the first place. Isolation technology makes this possible by enabling a Zero Trust approach to security. All web content — whether malicious or not — is treated as bad, and executed in an abstracted layer in the cloud. Every web page a worker visits, link they click on and email attachment they download happens on a virtual browser in the cloud, making it so potential threats like malware and ransomware never come anywhere near the endpoint. With isolation technology that enhances visibility into the browser and offers adaptive security policies, security teams can protect against HEAT attacks, suspicious website behaviors, and zero-hour threats in real time. Users are only ever exposed to sanitized, safe content, completely cutting off attackers' opportunity to gain initial access to the network. Because if a threat can't get in, it's not a threat.


# About **Menlo Security**

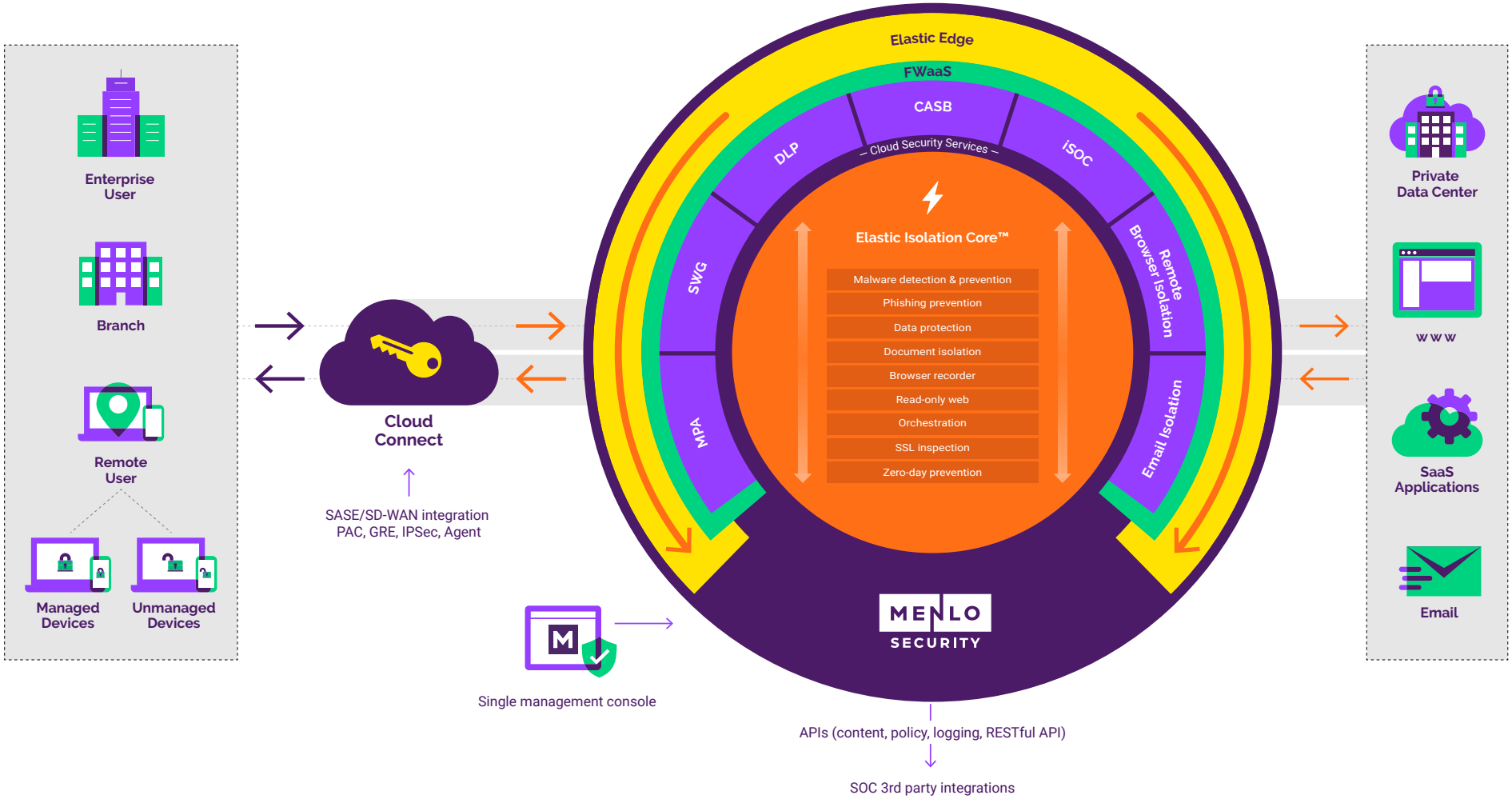
Menlo Security protects organizations from cyberattacks by eliminating the threat of malware from the web, documents, and email. It focuses on protecting the single biggest productivity driver for knowledge workers — the web browser.

Menlo's Cloud Security Platform prevents threats from entering an organization and secures data and application access in a single, global cloud-based offering. Our Elastic Isolation Core™ creates separation between the user, content and applications where security, policy and visibility are applied. By preventing threats before they happen, as opposed to detecting and responding, organizations eliminate all threats, including Highly Evasive Adaptive Threats (HEAT) across web, email, SaaS applications and private applications.

## HEATcheck

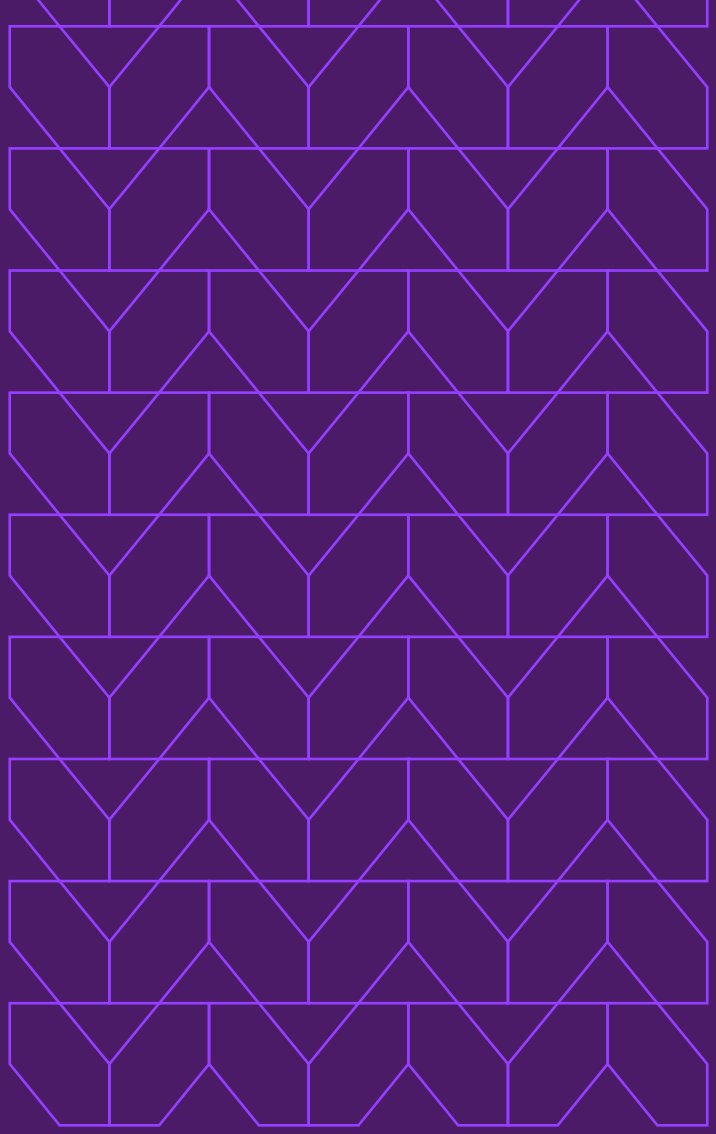
Menlo Security provides a lightweight penetration assessment to help organizations better understand any susceptibility to various HEAT attacks. The assessment leverages various real-world HEAT attacks currently being used by threat actors, safely allowing organizations to deduce their exposure. Menlo's HEAT Check tool does not deliver actual malicious content.

# Menlo's Cloud Security Platform



# Get in touch with us.

**Contact** us today to learn if your organization is currently susceptible to HEAT attacks, but most importantly, how you can make them never happen in the first place.



**menlosecurity.com/heatcheck**

ask@menlosecurity.com

MENLO SECURITY