




GRADIENT

Securing Digital Infrastructure Against the Next SolarWinds Attack

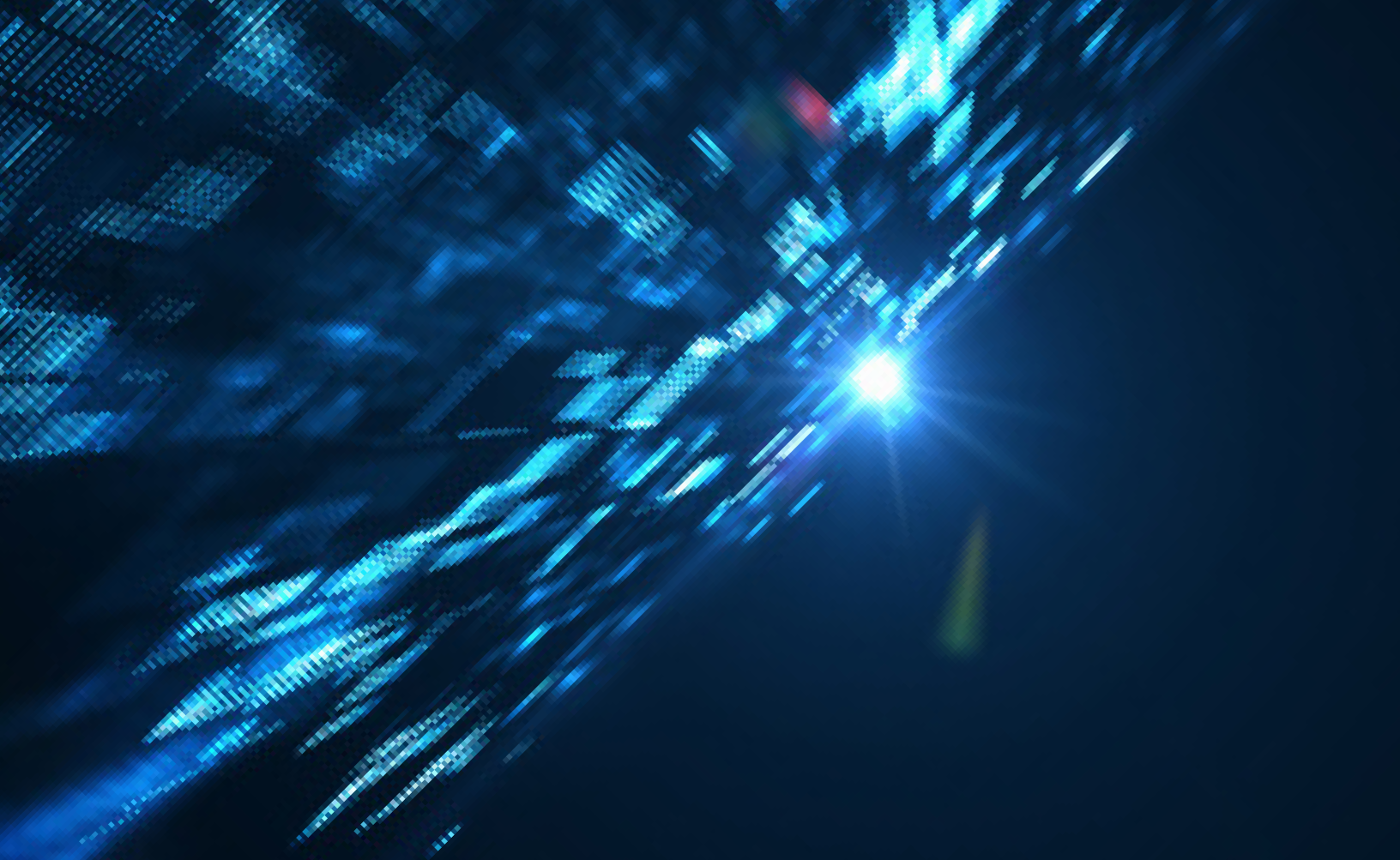
WHITE PAPER



The current cybersecurity mantra of detection and response (D&R) has failed to adequately protect us. These failures are often public, significant in scale, pervasive in scope, and can be financially disastrous to the organization. In response, we propose **Gradient Cybersecurity Mesh**: an upgrade from the currently reactive, fragmented Detection & Response paradigm to a **proactive, unified defense for all assets, users, and data.**

Contents

The Existential Cybersecurity Threat	4
Summary of the SolarWinds Attack: Scope and Approach	5
Why Is the SolarWinds Hack (Still) Such Cause for Concern?	6
Gradient's Solution: A Completely New Approach	7
Addressing Zero Trust	8
Analysis of Today's Vulnerabilities	9
Zero Trust User (+Device) Authentication	10
How Today's Best Fails	10
How SUNBURST Is Being Detected and Mitigated Today	18
Conclusion	19
Appendix: Brief Summary of How the SolarWinds Compromise Unfolded	20



The Existential Cybersecurity Threat

Few cybersecurity compromises have garnered the level of attention of the SolarWinds hack, revealed in December 2020, and perpetrated by Russian foreign intelligence service (SVR), a.k.a. APT29 or “Cozy Bear.” In this cyberattack the Russian agents managed to install software backdoors in at least 18,000 private sector and government organizations worldwide, enabling the attackers to take control of servers and exfiltrate data over a period of nearly nine months before being detected.

Alarming, more than a year later, we are not better positioned to prevent another attack of the scale or type of SolarWinds. This alone should give pause to private sector executives, corporate governance boards, and government leaders. But this concern is even more real in the face of the geopolitical conflict that continues to unfold at this moment between Russia and Ukraine, the US, and NATO allies.

In what follows, **we outline the specific vulnerabilities** exploited by the SolarWinds attack and other recent high profile cyberattacks that successfully took down sophisticated organizations despite best practices cybersecurity deployments.

We introduce the new approach and technologies of Gradient Cybersecurity Mesh, or more simply “GCM,” a platform capable of securing the core elements exploited in the SolarWinds compromise and others. We describe:

- 1.** How deployment of **GCM prevents such compromises in the first place,**
- 2.** How it **enables self-healing** of devices, software, and users, and,
- 3.** How it **delivers swift, secure remediation** of any compromised endpoints.

Summary of the SolarWinds Attack: Scope and Approach

Ignoring the time period when SolarWinds's own network was compromised, nearly nine months elapsed between the first customer network compromise and the discovery of the SolarWinds malware. In that time, publicly released figures indicate that the SolarWinds attack led to infiltration of approximately 18,000 organizations, including US government agencies and high-profile Fortune 500 companies. Critically, it appears the hackers decided several months prior to detection that they had achieved their goals, and, in order to hide their tracks, quietly removed the backdoor from SolarWinds's network that was used to release customer targeted SUNBURST malware. This begs the questions: How much broader would the scope of compromise have grown if the malware was left in place? How certain are we that all secondary malware was detected and mitigated?

The SolarWinds Attack summary:

- 9 months until discovery
- 18,000 organizations infiltrated
- US Government agencies and Fortune 500
- Perpetrators stopped on their own
- All known commercially available cybersecurity tools failed to detect or prevent the compromise

In terms of the attack itself, the perpetrators of the SolarWinds compromise leveraged vulnerabilities in the software supply chain to infiltrate the legitimate build environment of the SolarWinds Orion platform, a widely used IT performance monitoring tool. SUNSPOT malware was used to release a second malware, SUNBURST, which established a backdoor inside each of the estimated 18,000 organizations that utilized the Orion platform. This backdoor was ultimately used to exfiltrate sensitive information and otherwise harm target

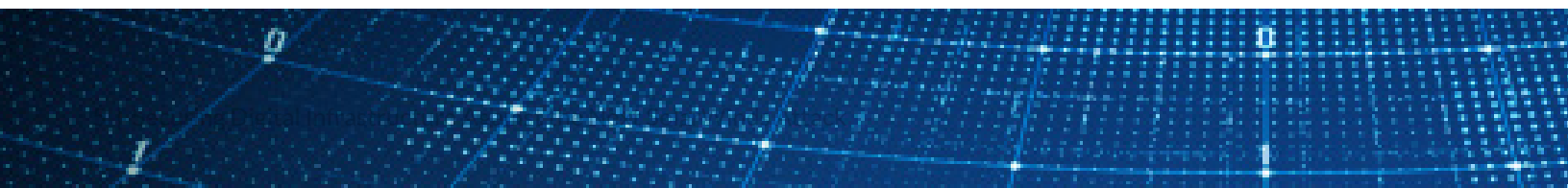
organizations including launching yet more malware. The full extent of damage is difficult to know.

What the attackers managed to do:

- Steal signing keys to remotely access the network through the "front door" of the authentication server
- Exfiltrate sensitive intellectual property assets
- Take control of machines, execute remote files, escalate privilege
- Disable system services, anti-malware & EDR tools
- Deploy subsequent malware
- Encrypt and obfuscate telemetry including exfiltrated data back to foreign C2 servers
- Move laterally through organizations undetected

Among other capabilities, the initial backdoor in the Orion platform, SUNBURST, enabled attackers to transfer files, execute files, reboot machines, and disable system services. We know that beyond SUNBURST, at least one additional malware was released as part of the attack, nicknamed SUNSHUTTLE. SUNSHUTTLE goes beyond SUNBURST to use encrypted HTTPS sessions with the command-and-control backend to avoid eavesdropping and better blend in with normal network traffic using popular referrer URLs, including facebook.com, google.com, and bing.com. This suggests that as yet undetected variants of malware may still be present.

In the wake of SolarWinds (2020-21), Triton (2018), Operation Aurora (2009), and other nation state sponsored attacks, **the US and UK intelligence communities have taken the position that organizations must assume they are compromised.** This is the impetus for us to build Gradient Cybersecurity Mesh.



Why is the SolarWinds Hack (Still) Such Cause for Concern?

In short, the SolarWinds hack is of existential concern because it exploited the most fundamental and sacred element used to convey trust in today's IT networks: the authentication credentials that communicate the legitimacy of an entity to access key resources, like an organization's payment infrastructure or sensitive databases. Instead of trying to brute force the authentication tools, which target organizations use to ensure that a given user or application is legitimate and has the authority to access a resource, the attacker simply went around them entirely. They did this by stealing the privileged credentials (the signing certificates, used to cryptographically endorse that this user was authorized) out from underneath these systems.

Security researchers were initially baffled in examining SUNBURST compromised networks - they didn't see the usual failed attempts to log in, any credential stuffing, any evidence of phishing, or other exploits of two factor authentication tools. What tipped them off to the mechanism of compromise was the lack of any attempts, valid or not, to authenticate. In fact, what had happened was that once in the network, the intruder used "the administrative permissions acquired through the on-premises compromise [the SUNBURST backdoor] to gain access to the organization's global administrator account and/or trusted SAML token signing certificate"¹. The hackers managed to exfiltrate the long-lived signing key associated with this signing certificate and used this to sign falsified authentication tokens directly. These false credentials were then used to access the organization's IT network, going through the metaphorical "front door" undetected.

Another reason this hack is still cause for existential concern is that it has highlighted the utter failure of current solutions. The state-of-the-art in cybersecurity detection and response tools, including the newest "XDR" frameworks, were completely

circumvented repeatedly, for months on end, in thousands of the most sophisticated organizations. In SolarWinds's own words, "by managing the intrusion through multiple servers based in the United States and mimicking legitimate network traffic, the attackers were able to circumvent threat detection techniques employed by both [sic] SolarWinds, other private companies, and the federal government."

While we've learned a bit more about how we might spot another attack once it's happened (e.g., yet more DNS servers and VPNs added to untrusted lists, new methods of traversing the network once inside characterized such that they may be used as heuristics for detection rules), we have not yet deployed defenses at scale to prevent these attacks. The same Russian SVR (a.k.a. "Cozy Bear") hackers believed to have launched SolarWinds as well as the Russian GRU (a.k.a. "Fancy Bear") hackers behind NotPetya, which caused an estimated \$10B in economic damage in 2017, are actively targeting Ukraine and NATO again.

Protecting the keys to the castle in the face of the utter failure of D&R:

- intruders used admin permissions acquired through backdoor
- got access to the trusted SAML token signing certificate
- used this to sign falsified authentication tokens directly
- utter failure of current threat detection and response solutions
- at thousands of the most sophisticated organizations

1. ["Customer Guidance on Recent Nation State Cyber Attacks," Microsoft Security Response Center, December 13, 2020](#)



Gradient's Solution: A Completely New Approach to Identity and Authorization

In the face of ongoing failures, a completely new approach to identity and authorization is needed. Gradient Cybersecurity Mesh is leveraging that approach and is built to prevent and protect against the SolarWinds Compromise.

Gradient's founding mission was based on the belief that the approaches taken to secure modern connected infrastructure are not enough, and that a wholesale re-imagining of how we do things is necessary.

The evolution towards conditional access will make the issuer of credentials the central target of compromise and requires extreme measures be taken to secure this validation server. To that end, we built the most secure processor on the planet from gate level up to be a robust defense to all known side-channel attacks.

Authentication credentials, because of their high value, must be made short-lived. Revocation lists are rarely used in practice and compromised credentials are the number one vector of network breach today. We architected GCM around ephemeral credentials issued for as short a period of time as practical – in some deployments these are even refreshed on a per hour basis.

Finally, GCM ensures these authentication credentials are granted only to valid users, using valid platforms, with the credential strongly bound uniquely to the platform; additionally, that remote attestation of a full stack security "fingerprint" must be performed regularly, and that credential issuance must be conditioned on successful verification. Depending on configured policy for the platform, any detected malware on an endpoint could automatically trigger an alert, result in a credential with reduced authorization rights, and/or isolate the endpoint through credential non-issuance (i.e., to "fail closed" rather than proliferate through a network).

These are the core attributes that make up GCM: a completely reimagined approach -

- Secure defenses for the verification server
- The most secure processor on the planet hosts sensitive operations
- Short-lived, ephemeral, automatically-updated credentials
- Credential binding only to authenticated users and platforms
- Remote attestation of complete asset fingerprint



Addressing Zero Trust

Even prior to the SolarWinds compromise, debriefs by intelligence officials on earlier cyberattacks had already been urging organizations to consider cybersecurity approaches that could remain operable and secure even in the event some elements of the network infrastructure had been breached by malware. These challenges drove the development of the so-called “Zero Trust” philosophy of security we see today. These challenges, and the failures of existing approaches to resolve them, drove us to build GCM.

We’re big fans of the philosophy of Zero Trust Architecture (ZTA) and believe, if implemented properly, ZTA can be transformational from a security perspective. Presidential Executive Order 14028 on May 2021 echoed the sentiment advocated by the US intelligence community that there is an existential need to establish ZTA as a basic security model, declaring that “[i]ncremental improvements will not give us the security we need.”

ZTA emphasizes that systems must verify that every device, user, or API is who they say they are, and has the permission to perform a given action, prior to granting access or authorization. Further, this should be time-limited access, especially if access includes escalated privilege.

The SolarWinds hack highlights just how critical it is to get the details right. It is a great lens through which to assess the shortcomings of most implementations of a “Zero

Trust Architecture” when it comes to this kind of Advanced Persistent Threat (APT).

To that end, we performed an analysis of the SolarWinds hack and other hacks, by the same and related APT groups, using data compiled from FireEye, Mandiant, Microsoft, Dragos, the MITRE ATT&CK database, and others, along with detailed code analysis. Based on this information, we identified the following four critical vulnerabilities in the current cybersecurity paradigm, even with adoption of a conventional Zero Trust Architecture:

Four critical components to get right with Zero Trust Architectures (and where most fail)

1. How to verify a device, user, or API (the “entity”) is trustworthy
2. What to verify about this entity to make this decision
3. How often to re-verify
4. And, nearly universally missing from the discussion: How secure is the verifier infrastructure itself?

Analysis of Today's Vulnerabilities

Our analysis assumes that the organization's endpoints and network as a whole have already adopted the absolute best practices. Highlights of these include:

A well-constructed ZTA implementation around identity:

- Strong identity-based verification of users, non-human users (e.g., APIs), and devices. In this setup, user identities are managed and enforced via an Identity Provider (IdP) such as Active Directory or Kerberos, along with second factor authentication (e.g., Duo, Okta, Ping).
- Every device has its own unique cryptographic identity, e.g., by using an existing managed Public Key Infrastructure (PKI) or Certificate Authority (CA) to issue credentials, e.g., digital certificates to every device.

Endpoints are secured by best practices:

- An agent-based endpoint detection and response tool (e.g., Microsoft Defender, SentinelOne) capable of providing:
 - Antivirus protection,
 - Monitoring of software processes,

- Monitoring of disk/memory access,
- Monitoring of network access,
- Remote management of upgrades and quarantine capabilities.
- The devices are running correctly configured UEFI versions to enable secure boot and upstream system protections, such as Linux IMA.

Network is secured by best practices, such as an XDR framework that includes:

- Agentless or dedicated agent-based network-wide monitoring,
- Monitoring of global events like CVEs (Common Vulnerabilities and Exposures),
- Network segmentation and virtualization, if applicable, such as VDI (Virtual Desktop Infrastructure) solutions accessed via "Zero Trust Application Access".



Zero Trust User (+Device) Authentication

In a typical scenario, prior to being granted access to a resource, e.g., a payments infrastructure or a customer database containing sensitive personal information, a user would be challenged to log in with username and password and a second factor authentication. To comply with the latest recommendations from the Office of Management and Budget (OMB) on Zero Trust Architecture, a device-level signal would also be queried, e.g., the access control system would also request the current security posture of the endpoint the user is utilizing to access (e.g., status from Defender agent). Other heuristics like IP address, geolocation, etc. may be included. If the Defender agent indicates healthy platform status, the username/password combination are correct, the user is valid according to the IdP, and the two factor authentication check passes, then the conditional access framework will use its signing certificate to endorse an access credential and

send this to the user+device entity. This user+device is now considered authenticated.

Having gone through this procedure, the user+device entity then presents this access token to the payments portal, which checks that this is indeed a valid authentication credential, and may verify secondarily, by passing the credential to an authorization service, that the user has authorization to access the resource. Following checks of both of these, the system grants the user access.

The rationale behind the described system seems straightforward and robust: surely, if the credential is valid, then the user or API must be legitimate, because it was checked by something trustworthy – and is therefore allowed access, right? Sadly, no.

How Today's Best Fails

Disconcertingly, the state-of-the-art cybersecurity infrastructure just described would not have stopped the SolarWinds attackers.

Why? SolarWinds offered the following explanation for why the attackers were able to persist inside the compromised network for so long:

“By managing the intrusion through multiple servers based in the United States and mimicking legitimate network traffic, the attackers were able to circumvent threat detection techniques employed by both (sic) SolarWinds, other private companies, and the federal government.”

Taking a fundamental look at the attack from the lens of the Four Critical Components to Get Right with Zero Trust Architectures, we can break down the failure more specifically:

1. Verify that the device, user, or API (the “entity”) is trustworthy

Current Point of Failure: Credential verification as done today doesn't go far enough, making it feasible to masquerade as a legitimate user with stolen credentials.

Just because I have a valid credential, (that is, legitimately signed by a valid authority, like SolarWinds's signing certificate), doesn't mean that the credential I'm presenting is actually mine, and not stolen from another intended user or API. There's nothing about this credential itself that uniquely associates it to me or to my device or my API. So, if I can steal this credential, I can masquerade as a legitimate user and perform the actions that user or API was granted under a typical role-based access control setup (RBAC). This type of credential exfiltration is exactly the first step taken in the SolarWinds compromise. Once past this step, today's systems

of detection and response, including XDR, are effectively hoping the attackers make a mistake such that behavioral detection algorithms detect the compromise. Consider that, in the case of the SolarWinds compromise, the detection and response paradigms of at least 18,000 organizations, including U.S. intelligence agencies, were successfully thwarted, in some cases for periods of 9 months or more.

60% of data breaches today occur because of stolen credentials

Credential compromise is bigger than SolarWinds: it's estimated that at least 60% of data breaches today occur because of stolen credentials. The WhisperGate malware launched against Ukraine's defense ministry in early 2022 leveraged compromised credentials as an initial entry point. Aside from direct exfiltration, other attack vectors like man-in-the-middle attacks, phishing attacks (including of second factor), and more can be mounted against the credential issuance and storage mechanisms used today.

Gradient Cybersecurity Mesh Innovation #1: Eliminating the Threat of Stolen Credentials

Gradient Cybersecurity Mesh makes it virtually impossible to steal credentials in the first place, by leveraging the hardware roots of trust already present on most enterprise laptops, desktops, servers, and cloud instances (e.g., Trusted Platform Modules (TPMs), Apple's T2 security coprocessor, Hardware Security Modules (HSMs), and Trusted Execution Environments (e.g., Intel SGX, AMD SEV)) to keep the sensitive cryptographic keys safe from hackers. Further, if desired, GCM can be configured such that one can inspect the credential itself to see that it was issued to a particular machine, or machine+user combination, enabling peer-to-peer level secondary enforcement.

2. Ensure that what you're verifying is meaningful and sufficient

Current Point of Failure: Current endpoint protection mechanisms like EDR do not go far enough to verify that a platform is trustworthy: for example, they do not verify that a platform's integrity is uncompromised and free from malware or memory-resident compromises. This failure arises because today's tools simply cannot see, let alone verify, the integrity of the lower level elements or memory subsystems. As a result, a malware infected machine can often be declared 'healthy', and granted access to the organization's network. Once allowed, attackers can move laterally and establish persistence, making it very hard to remove or even fully assess the scope of compromise.

Today's endpoint detection and response tools run as software that is launched by the operating system to monitor unusual activity, e.g., unusual file access patterns, network access patterns, or other behaviors that may be correlated with malicious software. These are valuable tools to catch known threats. But they are only as good as their preconfigured detection rules, and faithful enforcement of these rules is predicated on lower level software being uncompromised.

Each new threat like the SolarWinds malware (SUNBURST) is explicitly designed to circumvent the threat detection rules in place at the time. The attacker has an extremely unfair advantage of time, often months or more, to learn the exact detection rules and curate the hack to your environment or some industry-specific subsystem.

The fundamental problem is that you're asking a potentially compromised system to self-report whether it is OK. The very thing you're trusting this tool to do, though, is something it cannot be trusted to do alone.

Low level system firmware is critical to protect because it is responsible for setting the foundational elements of platform security. Once breached, any other cybersecurity measures, any data the platform contains, as well as any data accessible to it via credentials it stores, e.g., encrypted databases, are moot for that device. This attack surface is left unprotected by best-in-class cybersecurity protocols today.

Detection and Response is effectively a game of cat and mouse, and one that organizations cannot expect to win, especially for new threats.

As an example, the SolarWinds hack was possible in part because the original malware was able to install a malicious executable file on the virtual machine that was used to build product software releases, and this modification was not detected by the security features of the build environment. A comprehensive measurement check of the codebase on boot would have shown a change to the measurement of this software process. Such a comprehensive validation is possible with modern computing systems, and it is essential that we do so continually, and to predicate access on such validations. We must protect low level software components from compromise so that other defenses like EDR remain viable. We refer to the process that enables this secure remote measurement as 'remote attestation' - it is at the heart of what makes GCM possible.

Such a comprehensive validation is possible with modern computing systems, and it is essential that we do so continually.... We refer to this process as "remote attestation."

Gradient remote attestation ensures that the platform itself is trustworthy, making Gradient Cybersecurity Mesh a fundamental requirement for any tool like EDR to function properly. Microsoft estimates that 83% of organizations have

been victim to firmware compromise in the last 24 months.² This need to secure the low-level firmware is such an issue that the NSA has proposed computers be replaced every three years just to ensure that what low level firmware security exists is maintained by the vendor.³

Gradient is not alone in utilizing hardware-level cryptographic measurements of code, but only GCM enables remote verifiability of integrity. Code measurement is also leveraged, for example, in protection features like UEFI Secure Boot. But Secure Boot is only as useful as it is current: so it can effectively halt the boot process or enter safe mode if it detects suspicious changes. Without remote attestation, however, this static Secure Boot code cannot keep up with today's rapidly evolving threat landscape, and with the potential for Zero Day vulnerabilities to be discovered on legitimate code.

83%

of organizations have been victim to firmware compromise in the last 24 months

Here are four highly-publicized compromises from the leading hardware platforms:

- Apple T2 Security Chip (2020) --- Secure boot process compromised.⁴
- Nvidia Tegra processor (2018) --- Secure boot process compromised as a result of Zero Day vulnerability in Bluetooth stack.⁵
- AMD Secure Enclave Virtualization (SEV) hardware (2020, 2021) --- multiple vulnerabilities.^{6,7}
- Intel Software Guard Extension (SGX) architecture (2016, 2018, 2020,2021) --- multiple vulnerabilities led to signing key leakage, speculative execution attacks, and others.^{8,9,10}

In most, if not all, of these cases, it was impossible for the user to know if their machine was compromised, because the

2. Microsoft's Security Signals report (March 2021)

3. August 2019: NSA Cybersecurity Information PP-19-1017 - "Leveraging Modern Hardware Security Features: Trouble Beneath the Surface"

4. "Apple's T2 Security Chip Has an Unfixable Flaw," Wired (October 2022)

5. "Vulnerability Disclosure: Fusee Gelee" (April 2018)

6. "SEVerity: Code Injection Attacks Against Encrypted Virtual Machines" (CVE-2020-12967)

7. "undeSErVed trust: Exploiting Permutation-Agnostic Remote Attestation" (CVE-2021-26311)

8. "Intel SGX Explained"

9. Foreshadow (CVE-2018-3615), "Foreshadow-NG: Breaking the Virtual Memory Abstraction with Transient Out-of-Order Execution"

10. SmashEx (CVE-2021-0186), May 2021

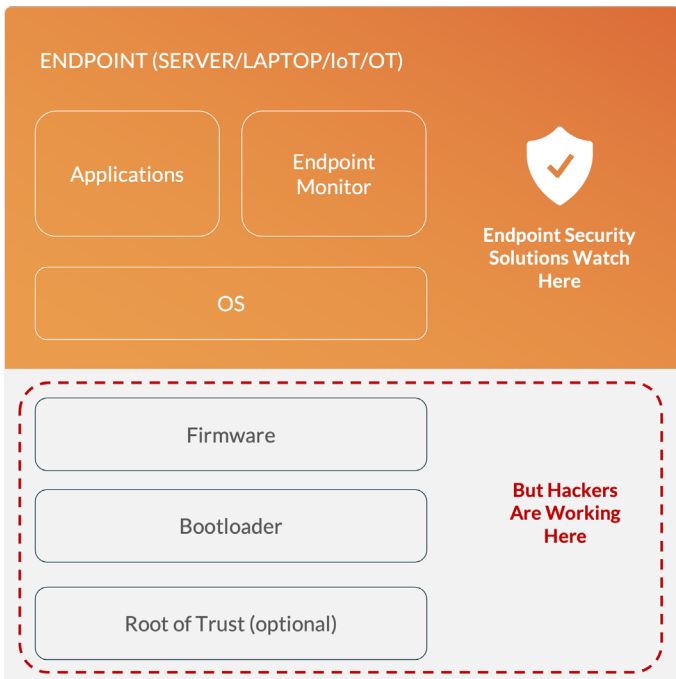
platform measurements were not exposed for third party validation. Remote attestation would have made the question of determining compromised device state trivial and enabled organizations to automatically isolate or deprecate permissions until patched.

Gradient’s Cybersecurity Mesh goes beyond all current approaches to:

1. Validate the entire software stack locally and remotely verify using the most secure verification processors in the world.
2. Evaluate complete user+device security posture continually, referencing to dynamic security policies that reflect up to the minute, global security environment.
3. Make this full stack security fingerprint visible to the other devices in your network, so that a truly “Zero Trust” verification is done by default, everywhere, without blind assumptions.

Now, this is “Attested Zero Trust.”

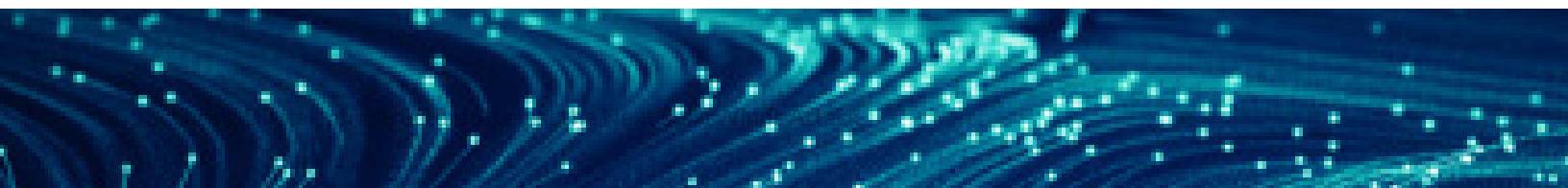
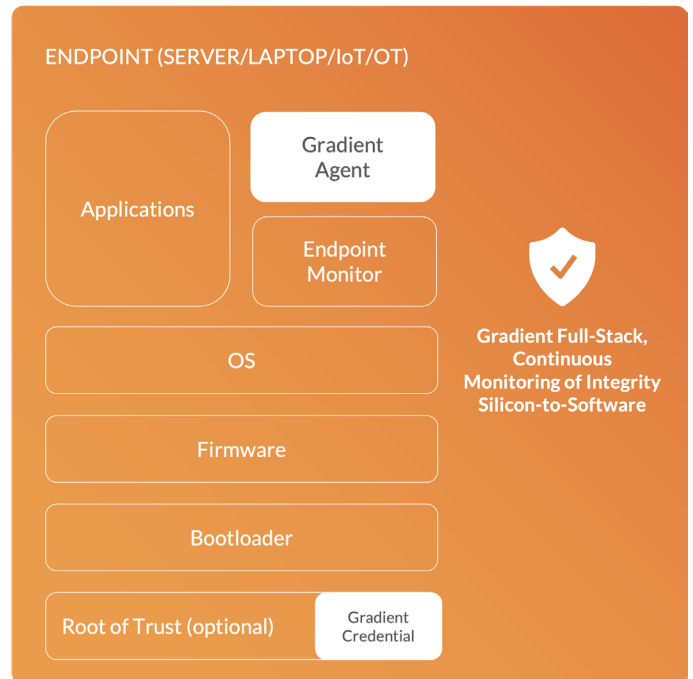
**Endpoint Monitoring:
Firmware Exposed**



**Gradient Cybersecurity Mesh Innovation #2:
Full Stack Attested Measurements**

Gradient Cybersecurity Mesh utilizes decentralized software agents to continually measure the full stack security “fingerprint”, and secure verifiers to check endpoints against centralized security policies to ensure that system software has integrity and is uncompromised by advanced malware - from the lowest levels of firmware up to the integrity of virtual machines, even memory itself when possible. GCM ensures that these full stack fingerprint definitions are synchronized across your organization, and that every device’s fingerprint is inspectable, so that a truly “Zero Trust” verification is done by default, everywhere, without assumptions.

**Gradient Cybersecurity Mesh:
Full-Stack Continual Integrity Verification**





3. Re-verify faster than threats evolve

Current Point of Failure: Digital infrastructure as it exists today struggles to keep pace with the dynamic, ephemeral nature of security, in part because of the overhead in renewing credentials. As a result, most systems use long-lived access credentials and keys that are valid for far longer than it would take an attacker to compromise them. Yet these endpoints still operate with significant privileges and access.

The concepts of automated just-in-time authorization and enforcement of principles of least privilege are beginning to take hold in commercially available platforms. We believe these concepts must extend further to things like the public key infrastructures (PKI) and certificate infrastructures prevalent in enterprises today, such that we can have confidence that anything with a valid credential has been recently checked for integrity, current user validity, etc. By making valuable credentials short-lived, we avoid the need to rely on cumbersome revocation lists that are rarely even used in practice.

In the case of SolarWinds, the organization-level compromise enabled by the SUNBURST backdoor was that the signing certificate for the SolarWinds Orion platform was exfiltrated and used to sign malicious tokens, which allowed the hacker to present legitimate looking credentials to access the network, as if they had been directly signed by the access control server. In addition to the other mitigations, if this signing certificate

had just been made short-lived, if the tokens it issued were made short-lived, or if the platform hosting the signing certificates had appropriately utilized secure hardware to protect them, then the initial compromise would likely have been prevented.

We expect objection to ephemeral credentials: “Won’t this break my infrastructure? Certificates are so hard to manage.” No! We agree that this is how things have been, but GCM automates away the burden of credential rotation. And, the trend line is clear that the world is moving with us. Large tech players like Apple and Google have pushed for shorter certificate lifetimes, with Apple not allowing more than 1 year certificates. Mozilla, Facebook, and Cloudflare have proposed short-lived, delegatable TLS credentials. Gradient Cybersecurity Mesh has been painstakingly engineered to work seamlessly within the bounds of the standard TLS stack and web browsers so that your organization can focus on core business drivers, and your IT and DevSecOps teams can be unburdened to focus on real threats, not a Lite-Brite of ambiguous alerts.

By moving to short-lived credentials, GCM shifts the balance of power in cyber warfare in your favor. We make it prohibitively expensive to compromise any given asset.

Gradient Cybersecurity Mesh Innovation #3: Secure Ephemeral Credentials (X.509, SSH keys, SAML tokens)

GCM enables your organization to move seamlessly from static, long-lived credentials to secure ephemeral credentials without breaking existing protocols like TLS. By moving to short-lived credentials, GCM shifts the balance of power in cyber warfare such that compromising an endpoint becomes prohibitively costly for the attacker.

This seamless switch to ephemeral credentials is achieved by plugging Gradient's secure verifier into your existing identity system (e.g., Certificate Authority (CA) or Public Key Infrastructure (PKI) to act as an intermediate CA, key, or token issuer), provided as a SaaS offering, on premises instance, or hybrid. Gradient Cybersecurity Mesh conditions credential renewal on a platform+user passing integrity checks - unifying platform integrity checking with identity credential issuance, and securing the identity perimeter by default.

To protect the conditional access verifier, the policy engine runs inside Gradient's Secure Enclave Processors, benchmarked as of 2021 as the world's most secure processor by the U.S. Department of Defense.

GCM also speaks WebAuthn and SAML protocols, and issues SSH keys, such that any token or credential format can be securely leveraged, across any platform or scale.

4. Secure the core identity system and conditional access verifier itself

Current Point of Failure: Multiple recent compromises including SolarWinds began with exfiltration of sensitive signing certificates that are used to endorse a credential as being legitimate and originating from the user's organization or trusted proxy. Systems generally don't utilize hardware-level security features to protect signing keys or credentials, either at the authentication server or individual endpoint. The foundation of trust in the network is literally stolen out from underneath an organization.

Explicitly, the SolarWinds hackers deployed malware in a .dll file in the software update released from SolarWinds to the host operating system of the Orion server inside the target organization. This file enabled multiple actions, including the launch of a process that established a backdoor to a domestic VPN, which successfully masqueraded command and control communications as legitimate network traffic, under the Orion Improvement Program (OIP) protocol, via HTTP to the C2 servers. In order to go undetected, reconnaissance results were hidden inside legitimate plugin configuration files. Through this backdoor, the hackers were able to exfiltrate the signing certificate and endorse malicious tokens as legitimate.¹¹

To prevent this compromise, systems must leverage roots of trust to secure the signing keys down to the hardware on the rightful host. This is in fact the precise guidance, following SolarWinds, that Microsoft released for its own Active Directory Federation Service (ADFS).¹² And yet cruelly, ADFS became the target of a similar compromise, named FoggyWeb, by the same hackers just months later, because their users did not act. Were this guidance heeded, the hackers would have had to actually present the falsified credential to the signing server to have them endorsed, likely triggering an anomaly detection tool if it were still active.

"What I cannot get is why customers still do not protect their keys... This was a key vector during the SolarWinds attack and the actor behind it is still chasing these keys."

- Microsoft Security Engineer (Fall 2021)

11. "Highly Evasive Attacker Leverages SolarWinds Supply Chain to Compromise Multiple Global Victims with SUNBURST Backdoor," FireEye (Dec 13, 2020)

12. FoggyWeb: Targeted NOBELIUM malware leads to persistent backdoor," Microsoft Threat Intelligence Center (September 27, 2021)



We have already described the multiple levels of baseline security leveraged or enforced by GCM. On every endpoint where possible, a hardware root of trust like a TPM is used to lock credentials to platforms and make it virtually impossible to steal them.

Because a Zero Trust Architecture makes compromises to any individual endpoint more costly, it would correspondingly make any weakness in the server(s) that actually enforce(s) access control the logical focus of attack. Given this, we believe it is critical to significantly harden the verifier operations as they will be increasingly targeted in the future.

After surveying commercially available processors and determining that no processor was immune to the myriad side-channel attacks and key leakage issues that plague

In the SolarWinds hack and many others, the foundation of trust in the network - the authentication token signing key - is literally stolen out from underneath an organization.

To prevent this compromise, systems must leverage roots of trust to secure the signing keys down to hardware on the rightful host.

modern speculative execution CPUs¹³ (analysis that would later be validated by Google and others¹⁴), the Gradient team concluded there were none available that could support our threat model: we've designed to secure against nation state sponsored Advanced Persistent Threat groups like those behind SolarWinds, NotPetya, Pipedream, WiperGate and others. We are guarding the keys to your castle and we take it seriously.

In 2018 we began a multi-year effort to build the world's most secure enclaved processor, the Gradient SEP. Designed from gate-level up, Gradient SEP is robust to all known side-channel attacks, formally verifiable, and, from a development standpoint, fully traceable. We use these in distributed network configurations to provide the kind of mission-critical, nation state robustness that we saw necessary to secure the Global 2000, but that we have made available to anyone. This is the heart of GCM.

The team that built the Gradient SEP includes the father of the secure processor, MIT professor Srinivas Devadas, as senior technical advisor to the project, his protégés as system architects, and chip industry veterans from Apple, IBM, and AMD. The immediate predecessor to Gradient SEP is MIT's Sanctum processor,^{15,16} which was vetted by the US Department of Defense using 13,000 hours of hacking

13. Example side-channel attacks include, e.g., Rowhammer, Meltdown, Spectre, Foreshadow, Foreshadow-NG, SmashEx, SEVerity, UndeSERVed Trust, etc.

14. See, e.g., <https://cloud.google.com/blog/products/gcp/protecting-against-the-new-l1tf-speculative-vulnerabilities> <https://googleprojectzero.blogspot.com/2018/01/reading-privileged-memory-with-side.html>

15. "Sanctum: Minimal hardware extensions for strong software isolation." (2015)

16. "Secure boot and remote attestation in the sanctum processor." (2018)

by 580 white hat hackers¹⁷ and found to have the fewest vulnerabilities of any design tested to date. We then spent the next 3+ years making it better (commercial ciphersuites like RSA and elliptic curve support, post-quantum cryptography support, and remote, post-quantum secure upgrade capability for the entire software stack including system cipher suites).

There are many reasons, beyond just side-channel immunity, that a hardened processor is critically important to protecting the identity system and beyond. Among these:

- 1. Processor Performance:** leveraging a Hardware Security Module (HSM) or TPM would not provide the level of performance we desired in terms of the throughput of credential rotations, in particular for RSA cryptography and future post-quantum cipher suites.
- 2. Security from the Ground Up:** TPMs and HSMs are trustworthy for the purposes of ensuring that cryptographic operations are performed with integrity and that keys are not easily exfiltrated. Things like security policies however require that the entire software environment is secure. This requires an “active” root of trust - namely, one that maintains control over the code loaded to the CPU. Google and Microsoft each came to the same conclusion at about the same time (2018), unveiling Titan¹⁸ and Pluton¹⁹, respectively, which have in common a custom silicon processor that resides on the motherboard of the server chassis to perform secure operations including key orchestration and inspection of code to the CPU. Gradient’s solution goes one step beyond Google’s Titan in that the secure processor is not just a supporting element but the host as well.
- 3. Crypto-Agility for a Post-Quantum Future:** we see crypto-agility as a necessary capability of secure hardware - that is, for the ability, while already field deployed, to securely upgrade the cryptography used to secure a network. As a concrete example of this, NSA guidance makes clear they expect public key cryptography to be vulnerable to quantum computers within the next decade. And yet, a post-quantum cipher suite has not yet been validated (although NIST is working on it).

So, we found ourselves having to design a system that must be viable for more than the five year time horizon of post-quantum needs, without the post- quantum solution actually available. Gradient’s Cybersecurity Mesh makes this kind of upgrade possible for endpoints as well as our core infrastructure, with our SEP’s full programmability as a 64-bit processor combined with our crypto-agile bootloader deployable to endpoints. In contrast, conventional HSM or TPM approaches will need to be retrofitted in the field with new hardware, costing an anticipated hundreds of billions of dollars globally.

Gradient Cybersecurity Mesh Innovation #4: Remotely Attested Secure Enclave Processors: The Most Secure in the World

To ensure verification operations are immune to compromise, highly available, and fault tolerant, Gradient utilizes our custom Secure Enclave Processors (SEPs) to run the software that enforces the policies that describe what attested security fingerprints correspond to legitimate devices, software, and users.

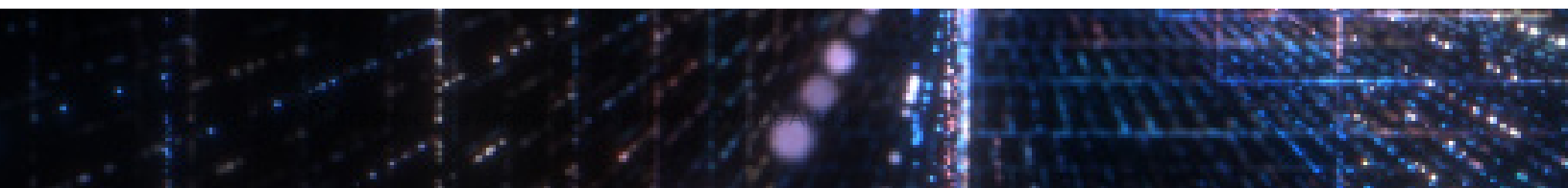
Functionally, these SEPs are hosted in AWS data centers and elsewhere, or can be provided as plug-and-play, on-premises 1U rackmount secure appliances, or hybridized. This federated deployment model enables high availability such that any changes to the global security environment can be rapidly propagated into the authentication status of every endpoint, user, and API in your organization.

SEPs also secure the code running the TLS protocols, token parsers, and any other sensitive cryptographic operations, and provide hardware accelerated key generation to enable your organization to scale to millions of credentials or more per day with a single hardware instance.

17. <https://spectrum.ieee.org/darpa-hacks-its-secure-hardware-fends-off-most-attacks>

18. [Titan: enabling a transparent silicon root of trust for Cloud, HotChips 2018](#)

19. [The hardware security platform behind Azure Sphere, HotChips 2018](#)





How SUNBURST Is Being Detected and Mitigated Today

In their analysis Mandiant notes that once the attacker gained access to the network with compromised credentials, they moved laterally with multiple different credentials distinct from the initial remote access, and that this “one to many” relationship between source system(s) and accounts can be used as an indicator of compromise. In normal business

operations one does not typically see a single system authenticating to multiple systems with multiple accounts.

This could be a detection and response pattern match after the fact. Better though, a Gradient secured organization would have locked out this malicious remote access in the first place, due to it not being a valid user on a valid, uncompromised machine.

Conclusion

The current detection and response approaches to cybersecurity continue to fail miserably. The failures are often public, significant in scale, pervasive in scope, and financially disastrous to the organization, whether due to direct losses or liabilities. Policy makers and cybersecurity insurance brokers have taken notice. In many cases premiums are drastically increasing for cybersecurity coverage unless proactive threat mitigations are in place. Gradient has completely re-imagined the approach to securing users, assets, and data as the new security perimeter, precisely to counter this trend of cyber compromises.

Gradient's Cybersecurity Mesh Solution, currently being deployed at flagship organizations across various sectors, introduces multiple unique innovations to address each of the failure points in current approaches, as illustrated by the highly sophisticated SolarWinds compromise:

- 1. Eliminating the Threat of Stolen Credentials:** Gradient Cybersecurity Mesh makes it virtually impossible to steal credentials in the first place, by leveraging the hardware roots of trust already present on most enterprise endpoints, immutably binding the user and device.
- 2. Ensuring Full-Stack, Ongoing, Attested Measurements:** Gradient Cybersecurity Mesh continually measures the full stack security "fingerprint" of every endpoint against dynamically configurable policies. Gradient's Neo's secure verifier is in turn powered by Gradient's Secure Enclave Processor, benchmarked as of 2021 as the world's most secure processor by the DoD.
- 3. Enabling Scalable, Secure, Ephemeral Credentials:** Gradient Cybersecurity Mesh enables a network to move seamlessly from static, long-lived credentials to secure, ephemeral credentials without breaking existing protocols like TLS.
- 4. Securing the Entire Stack Down to the Silicon:** Gradient Cybersecurity Mesh is enabled by its own remotely attested Secure Enclave Processors: the most secure in the world and ready for the post-quantum world.

**Is your organization ready to go beyond fighting the losing battle of Detection and Response?
Are you ready to eliminate the persistent, ongoing sources of high-risk compromise?**

Stop searching for the best-in-class in a class that is compromised by default. [Get in touch](#) and find out more about a completely new and comprehensive class of solutions.

APPENDIX: Brief Summary of How the SolarWinds Compromise Unfolded

The initial discovery of the SolarWinds compromise is credited to security firm FireEye, who uncovered a backdoor in the SolarWinds Orion software tool resident on their own network, during investigation of the theft of their Red Team toolkit by a nation state actor, between December 8 and 13, 2020.

In analyzing SolarWinds's own systems, it was discovered that the virtual machine-based build environment used to create Orion platform software updates had been compromised nearly a year earlier, using a malware referred to as SUNSPOT. Hackers appeared to have gained access to SolarWinds networks as early as September 4, 2019 and began testing the ability to release versions of SUNBURST, the malware that would insert a backdoor into the Orion platform in end-customer networks.

SUNSPOT turns out to be an executable file (taskhostsvc.exe) that maintains persistence by creating a scheduled task to execute whenever the host boots.

SUNSPOT first grants itself debugging privileges by modifying its own security token to add SeDebugPrivilege. From here, it begins to read other software processes in memory to identify when the VMware virtual machine that runs the build process is about to compile code.

Specifically, SUNSPOT monitors MsBuild.exe processes, part of MS Visual Studio, looking for any process that could be associated with initialization of the Orion build tools, and, if so, hijacks this build operation to inject SUNBURST.

The end result of this process is a DLL file, SolarWinds.Orion.Core.BusinessLayer.dll, that, while being legitimately signed by SolarWinds and released into the wild, is in fact a trojan horse hiding a malware payload known as SUNBURST.

Evidence shows the initial compromised code release, signed and released by SolarWinds as a software update to customers of the Orion Platform, began on March 26, 2020, and that this compromise remained resident inside SolarWinds until June 4, 2020, when it was removed by the threat actor, presumably to cover their tracks.

Once installed, SUNBURST lays dormant for a period of approximately two weeks before activating itself, ultimately opening up a persistent backdoor between the target organization and a domestic VPN host used by the attacker as command-and-control infrastructure. SUNBURST was able to hide this traffic inside the otherwise legitimate HTTP traffic of the Orion Improvement Protocol (OIP), thus evading detection even while actively exfiltrating information.

The SUNBURST backdoor allows the compromised code to transfer files, execute files, reboot machines, and disable system services. Network traffic masquerades under the legitimate Orion Improvement Program (OIP) protocol to communicate via HTTP to third party servers. In order to go undetected, reconnaissance results are hidden inside legitimate plugin configuration files.

Using the MITRE ATT&CK Framework, SUNSPOT leveraged the following TTPs:

Reconnaissance	T1592.002 Gather Victim Host Information – Software	StellarParticle had an understanding of the Orion build chain before SUNSPOT was developed to tamper with it.
Resource Development	T1587.001 Develop Capabilities – Malware	SUNSPOT was weaponized to specifically target the Orion build to replace one source code file and include the SUNBURST backdoor.
Persistence	T1053.005 Scheduled Task	SUNSPOT is persisting in a scheduled task set to execute after the host has booted.
Defense Evasion	T1140 Deobfuscate/Decode Information	The configuration in SUNSPOT is encrypted using AES128-CBC. It contains the replacement source code, the targeted Visual Studio solution file name, and targeted source code file paths relative to the solution directory.
	T1027 Obfuscated Files or Information	The log file SUNSPOT is encrypted using RC4.
	T1480 Execution Guardrails	The replacement of source code is done only if the MD5 checksums of both the original source code file and backdoored replacement source code match hardcoded values.
	T1036 Masquerading	SUNSPOT masquerades as a legitimate Windows Binary, and writes its logs in a fake VMWare log file.
Discovery	T1057 Process Discovery	SUNSPOT monitors running processes looking for instances of MsBuild.exe.
Impact	T1565.001 Data Manipulation Stored – Data Manipulation	Modification of the Orion source code to inject SUNBURST.

What is Gradient

Gradient offers the only cybersecurity solution that continually protects and communicates, via patented secure hardware attestation, the complete security posture of every platform, all the way from the legitimacy of the hardware to the firmware (UEFI), kernel, kernel packages, and more, to establish a dynamic “fingerprint.”

Gradient enhances the conventional authentication and conditional access flow for users, devices, and APIs to include the continual validation of both identity and the complete platform fingerprint. As a result, Gradient ensures that only legitimate users on valid, legitimate machines running correct, uncompromised software are allowed, where each of these attributes is re-evaluated at regular intervals to ensure they reflect the most up-to-date information on the state of every device on your network. This is dynamic attribute-based access control (ABAC) for everything, everywhere.

