Questions for the Record
Senate Select Committee on Intelligence
Hearing on the Hack of U.S. Networks by a Foreign Adversary
February 23, 2021

Questions for the Record for Mr. Brad Smith, President of Microsoft

[From Chairman Warner and Vice Chairman Rubio]

Regarding the ability to misuse identity access tokens described by Senator Wyden:

1. **When did Microsoft first learn of this vulnerability?**

Microsoft, together with other companies, governments, and cybersecurity professionals, first learned of the Golden SAML (Security Assertion Markup Language) post-exploit technique in 2017 when security researchers from CyberArc published[1] a public blog detailing the theory. Their research concluded that it was possible for attackers to "gain access to any application that supports SAML authentication (e.g., Azure, AWS, vSphere, etc.) with any privilege they desire…". Their research further noted that if attackers gained privileges in one environment, such as an on-premises active directory, then they could abuse that access to generate SAML tokens, which could open access to a victim's cloud environment.

In only 15 percent of the recent attacks we saw, the actor used the Golden SAML technique to access Office 365 (O365) accounts. The identity access tokens the actor was able to generate worked as intended – because the actor had first obtained the ability to act as a network administrator in the network and could issue tokens trusted by O365. There was no vulnerability in any Microsoft product or service that was exploited, but the inherent weakness in the SAML industry standard authentication system described in the CyberArc research did provide the actor with this method – among many others – to access other network resources.

Although Office 365 customer data was accessed as a consequence of this attack, we have found no indication that Active Directory (AD) was a vector in this attack – nor that any vulnerabilities in AD were leveraged. There has been speculation that a flaw in AD allowed users to gain elevated access. That is not accurate; the threat actors in this attack used several techniques to escalate privileges and/or obtain privileged credentials that gave them the ability in on-premises networks to act like a system administrator. Our investigations have confirmed several compromise techniques, including password spraying and spear phishing, which enabled the actors to obtain privileged credentials in a customer's environment.

The most advanced threat actors operate with ample patience, time, and resources. In this case, one of the things that the threat actor did extremely well was to appear to stay within the bounds

[1] Golden SAML: Newly Discovered Attack Technique Forges Authentication to Cloud Apps (cyberark.com)

of how a product or service should be used. The threat actor did not discover or use a new vulnerability in any cloud services that we have seen; rather, attackers took advantage of how a service was implemented in a customer's environment and then leveraged how the service worked to conduct their operation.

2. What steps did Microsoft take to protect against its misuse?

Microsoft advised customers to implement an expanding array of best practices that would help protect against the misuse relating to the Golden SAML post-exploit technique. It's important to emphasize again that use of the Golden SAML technique in the SolarWinds attacks took place *after* an attacker had first obtained the ability to act as a network administrator. Especially given a myriad of potential cybersecurity priorities, the most effective course of action was to prevent an attacker from establishing this ability in the first place.

Microsoft has emphasized five best practices, in particular:

1. Utilize multi-factor authentication;[2]
2. Establish Least Privileged Access principles across your network, and adopt other Zero Trust Principles;[3]
3. Secure the most sensitive credentials of network administrators and others in hardware or in the cloud;
4. Secure devices through Intune;[4] and
5. Use anti-malware tools such as Defender.[5]

We also continued to encourage our customers to move to newer authentication technology, including by moving their authentication practices to the cloud.

The Golden SAML theory that the CyberArc researchers presented would first require an attacker to already have gained a presence in a victim's on-premises network. The attacker would then need to acquire credentials of a network administrator in some manner. Following these steps, the Golden SAML technique could then be executed.

Microsoft's security researchers reviewed this theory, as they do with the thousands of possible attack vectors researchers regularly surface. Given the level of access an attacker would already need to have achieved prior to leveraging this particular technique, coupled with the reality that the initial attack vectors used to gain this level of access were active threats against customers,

---

[2] How to implement Multi-Factor Authentication (MFA) - Microsoft Security; Multi-Factor Authentication (MFA) - Microsoft Security; Set up your Microsoft 365 sign-in for multi-factor authentication

[3] Zero Trust Deployment Center | Microsoft Docs; Zero Trust - Microsoft Security; Using Zero Trust principles to protect against sophisticated attacks like Solorigate - Microsoft Security

[4] What is Microsoft Intune - Azure | Microsoft Docs

[5] Next-generation protection | Microsoft Docs

we prioritized protecting our customers against the initial attack vectors that would lead to multiple potential subsequent threats.

Additionally, our experience shows us that CISOs need to prioritize how they deploy their resources, and even when information about notional post-exploitation techniques is available, combatting such techniques will not necessarily be considered a high enough risk to divert time or resources from other, potentially more important, security efforts.

It's also worth noting that SAML is an old standardized technology that was first introduced in 2002 and subsequently updated in 2005 with SAML 2.0.[6] Given the age of SAML and the limitations of the protocol, we have long encouraged our customers to move to modern authentication technology that is more secure, specifically OAUTH2[7], which we also support in our products. However, many of our customers still require or opt to use SAML, often due to legacy applications that rely upon SAML authentication, making it challenging for them to change. For that reason, cloud providers continue to support SAML authentication, despite its limitations.

We have certainly reflected on what we can do in the future to ensure our customers are better prepared and that we are providing them with the tools they need to better protect themselves. That is why we announced recently that, for the next year, we will make available to government customers our advanced logging capabilities for no cost,[8] as we continue to review our efforts to support customer security. It is also why we continue to encourage organizations to at least move their authentication systems to the cloud. The attack on SolarWinds and its customers shows firsthand how challenging it is for organizations to protect themselves on-premises in today's cyber environment. Regardless of platform or tool leveraged, moving authentication to the cloud is one of the most impactful steps an organization can take to protect itself and its infrastructure.

[From Senator Wyden]

During your opening remarks, you stressed the importance of software security updates and stated that, when hackers tamper with the software updating process, it puts the entire world at greater risk.

3. **Have hackers ever compromised any of the digital infrastructure that Microsoft uses to create, authenticate, and distribute software security updates? If yes, please detail each incident, its impact, and whether Microsoft reported it to the appropriate U.S. government authorities.**

To our knowledge, Microsoft has never been the victim of a successful attack on our software security update channel. In 2012, Kaspersky and CrySys Lab found malware they called "Flame," which impersonated the Windows update channel by appearing to come from

[6] What is SAML? How it works and how it enables SSO | CSO Online
[7] An Introduction to OAuth 2 | DigitalOcean
[8] Addressing Audit Log Storage for U.S. Federal Government Customers - Microsoft in Business Blogs

Microsoft—but it did not involve a compromise of our update channel; it involved an imposter. As a result of what we learned from that event, however, we spent significant resources to harden our update infrastructure to prevent an actual compromise from occurring, and we continue to maintain vigilance today.

4. **What security tools or features are included with the E5/G5 license, which is Microsoft's most expensive enterprise software subscription, but not the standard E3/G3 license that might have aided in the discovery or mitigation of the identity compromise at issue in many of the Solorigate compromises or the operation of any subsequent deployed malicious software?**

As explained below, Microsoft's G3/E3 license includes *core* solutions for security, compliance, identity, and management, and the G5/E5 license includes *advanced* solutions. This provides customers with the ability to choose what they want to procure from Microsoft as well as a wide variety of other cybersecurity vendors. And even more important, the effective implementation of the core solutions in the E3/G3 license would absolutely have aided in the discovery or mitigation of the identity compromise at issue in many of the Solorigate compromises, as well as the operation of any subsequently deployed malicious software.

It's worth recognizing that cybersecurity is at an inflection point as threat attack sophistication escalates, digital attack surfaces increase exponentially, and customers navigate different economic constraints and talent scarcity. This environment requires a Zero Trust, multi-tiered security strategy, involving comprehensive protection as well as choice and flexibility based on business requirements and security solutions that organizations may deploy from multiple vendors.

Microsoft's approach to security is anchored in Zero Trust principles, and we provide defense-in-depth security across all layers, from development to operational security and in our baseline security capabilities that are included in our platforms by default. All of our customers benefit from baseline security capabilities, such as our Security Development Lifecycle (SDL),[9] Operational Security Assurance (OSA)[10] practices, Windows Defender Antivirus (AV),[11] Azure Security Center,[12] and Audit Logs for 90 days[13] by default.

In addition, we provide "core" (G3/E3) and "advanced" (G5E5) solutions across security, compliance, and identity management for customers who seek choice and flexibility in the highly fragmented security market, where they procure security solutions from more than 70 different vendors on average. Microsoft security solutions are built to address our customers' requirements and circumstances. We offer multi-platform (i.e., we also support Android, Linux, and iOS) and

[9] About the Microsoft Security Development Lifecycle
[10] Microsoft Operational Security Assurance Practices
[11] Next-generation protection | Microsoft Docs
[12] Azure Security Center | Microsoft Azure
[13] Manage audit log retention policies - Microsoft 365 Compliance | Microsoft Docs

multi-cloud (i.e., we also support AWS and Google Cloud), best-of-breed and best-of-integration solutions.

To deliver the best security protection from baseline to advanced, we fiercely innovate, and that requires us to invest across our portfolio of people and technology. Microsoft annually invests more than $1 billion in R&D and security operations, with more than 3,500 people working in security.

CORE (G3/E3) and ADVANCED (G5/E5) SECURITY in Microsoft 365

Most of our enterprise and public sector customers are seeking advanced security and choice in a highly fragmented security market segment with thousands of vendors. Our customers want best of breed as well as best of integration. They want multi-platform and multi-cloud alongside cost savings. Microsoft advanced security was built for customers based on their requests and requirements – to offer them more choice and protection on their terms, customized for their use cases and requirements, depending on what they already own and use.

G3/E3 includes *core* solutions for security, compliance, identity, and management, and G5/E5 includes *advanced* solutions. This allows customers to choose which capabilities are the best fit for their needs at different price points, and it also crucially enables flexibility to purchase and use security products from different vendors consistent with their broader strategies. The ability of our solutions to mitigate any threat or incident response scenario, such as the discovery or mitigation of an identity compromise or the operation of malicious software, depends in part on customer implementation. However, effective use of our core solutions would have aided in the discovery or mitigation of threat activities relevant to the Solorigate compromise.

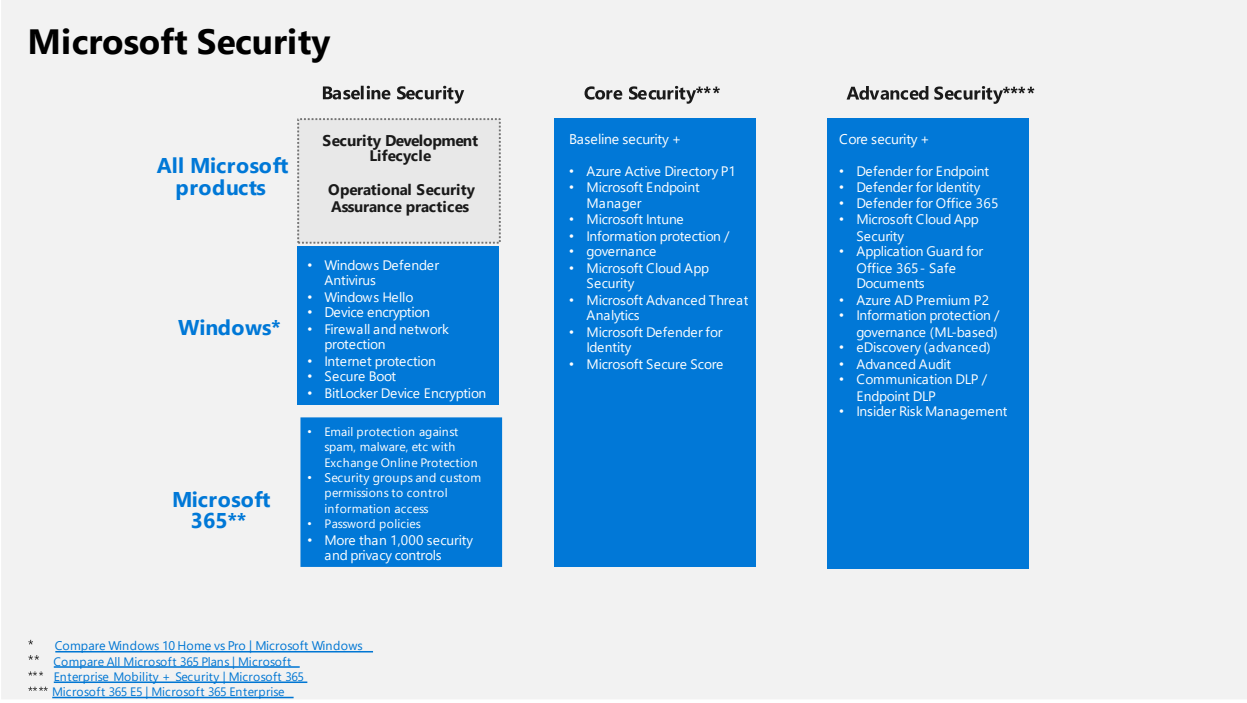
Core (E3/G3) Solutions

- Endpoint and App Management: Microsoft Intune, Mobile Device Management, Mobile Application Management.
- Threat Protection: Defender AV, Defender Firewall, Defender Exploit Guard and Credential Guard, Bitlocker and Bitlocker to Go, Windows Information Protection (these are all part of Windows E3), Desktop Analytics.
- Identity and Access Management: Azure AD Premium – that includes Conditional Access and Cloud App Security Discovery.
- Information governance, information protection and eDiscovery: including manual retention labels, basic policies, sensitivity labels, data loss prevention, office message encryption, content search, litigation hold and basic audit (e.g., out of the box protections such as encrypt-only or do-not-forward policies to sensitive emails with Office 365 Message Encryption).
- Unified Audit Logging (UAL): for M365 E3/G3 and O365 E3 customers, we store Basic Logs for 90 days by default, which provides organizations with visibility into many types of audited activities across many different services in Microsoft 365. Via

the Office 365 Management Activity API, customers can download and store the audit log data on their own for as long as they'd like.
    - For customers who need to retain audit logs for more time, we offer Advanced Audit in our comprehensive E5 offering or via the add-on E5/G5 Compliance SKU (and M365 G5 eDiscovery and Audit), which includes Advanced Audit, and allows customers to store audit logs for up to one year and provides additional capabilities for investigation.
- Microsoft Secure Score: helps customers assess their current state of security posture, improve said posture, and compare with benchmarks.

Advanced (E5/G5)

- Threat Protection: Defender for Endpoint, Defender for Identity, Defender for Office 365, Application Guard for Office 365, Safe Documents.
- Identity and Access Management: Azure AD Premium P2 – including risk-based Conditional Access, Privileged Identity Management, Access Reviews, and Entitlement Management.
- Cloud Security: Microsoft Cloud App Security.
- Information governance, information protection and eDiscovery: rules-based and ML-based retention labels and policies, records management, automatic and ML-based sensitivity labels, communication DLP, endpoint DLP, advanced Office message encryption, Insider Risk Management.
- Advanced Audit: included for M365 G5/E5 and O365 E5 customers, builds on UAL and retains all Exchange, SharePoint, and Azure Active Directory audit records for up to one year (for M365 E5/G5 and O365 E5 customers, we store Unified Audit Logs for 90 days by default, which provides organizations with visibility into many types of audited activities across many different services in Microsoft 365). Retaining audit records for longer periods can help organizations to conduct forensic and compliance investigations by increasing audit log retention required to conduct an investigation, providing access to crucial events that help determine scope of compromise, and faster access to Office 365 Management Activity API**.** Audit and retention needs are different by customer depending on industry and whether they are using 3rd party solutions. This approach gives customers choice and flexibility on what they pay for in the product, given the storage costs to retain.
    - We also offer customers the option to pay for further retention/storage up to 10 years via different plans based on their needs.



In a blog post published in 2015, a senior Microsoft employee wrote that:

> ***"The token signing certificate is considered the bedrock of security in regards to ADFS [Active Directory Federation Services]. If someone were to get ahold of this certificate, they could easily impersonate your ADFS server."***

In 2017, a researcher demonstrated that these encryption keys could be stolen and used to create tokens that could then be used to log in to accounts with cloud service providers, such as Office 365. The researcher dubbed this attack "Golden SAML." This hacking technique was then exploited by the adversary in the Solorigate incident. Microsoft has confirmed to my office that the company did not warn the U.S. government about the Golden SAML attack because it had not seen hackers exploit it in the wild.

5. **As a general rule, does Microsoft only warn its customers about vulnerabilities that adversaries are exploiting in the wild? If not, please explain why Microsoft chose not to warn its customers about this hacking technique but has opted to warn its customers about other vulnerabilities before they were actively exploited in the wild.**

Microsoft warns customers about hacking techniques in a wide variety of circumstances. It's worth noting once again that the Golden SAML theory became known to cybersecurity professionals at Microsoft and across the U.S. Government and the tech sector at precisely the same time, when it was published in a public paper in 2017.

Microsoft's warnings to customers are based in part on adherence to the principles of Coordinated Vulnerability Disclosure ("CVD"), the goal of which is to provide timely and consistent guidance to customers about vulnerabilities and other issues to help them protect themselves.[14] External researchers or internal Microsoft employees may discover a vulnerability in Microsoft products or services. Under CVD principles, information about the vulnerability is typically not released until an update to address the vulnerability is released. In some instances, even before an update is ready but when it's discovered that attacks using the vulnerability are underway in the wild, we disclose vulnerability information in coordination with the discoverer as appropriate under the circumstances to protect customers.

Again, the hacking technique identified in 2017 as Golden SAML did not involve a vulnerability in Microsoft products or services. The technique was one that demonstrated how an attacker with escalated privileges in a network could take advantage of an industry standard authentication process, SAML, to access other resources. Given that the attack technique did not involve such a vulnerability, the researcher did not report it to Microsoft under CVD principles. Instead, the researcher made the information about the hacking technique publicly available, so that it was equally available to U.S. government agencies, Microsoft and other industry participants.

In some cases, Microsoft shares our own research or amplifies publicly available information about hacking techniques or security priorities. Our customer communication has for years focused on core cybersecurity hygiene that, if properly deployed, would have protected against or minimized the impact of the attack on SolarWinds and its customers. As an industry, we still have work to do to get customers to take basic cybersecurity hygiene steps, which help to defend against and limit the impact of nearly all cybersecurity attacks. As the question notes, prior to the attack on SolarWinds and its customers (in which the Golden SAML technique was not the primary exploit used), there was no evidence that the Golden SAML technique had previously been used in an actual attack in the wild. Additionally, the Golden SAML exploit was not prioritized by the intelligence community as a risk, nor was it flagged by civilian agencies or other entities in the security community as a risk that should be elevated over promoting MFA deployment, combatting ransomware, or undertaking other fundamental security actions. Our experience shows us that CISOs prioritize how they deploy their resources, and even though information about this notional attack vector was available, addressing it was not considered a high enough risk priority to divert time or resources from other potentially more important security efforts.

Customer security is a core priority of ours, and we are constantly reflecting on what we can do better to create a more secure ecosystem. As noted in the question, we prioritize warning customers of vulnerabilities when they are identified and we have a patch or appropriate path to mitigation to share. The Golden SAML technique is different, not only because it was not previously exploited in the wild, but also because it is not a vulnerability but rather a post-exploit attack technique. To use this technique requires an attacker to successfully compromise a network, and then successfully acquire escalated privileges. At that point, as the data from this

[14] https://www.microsoft.com/en-us/msrc/cvd; The CERT Guide to Coordinated Vulnerability Disclosure (cmu.edu)

incident demonstrates, many other approaches to identity compromise are available to an attacker – approaches used by this attacker 85 percent of the time. Prioritizing defenses to this one post-compromise SAML token exploit would not be a good use of an enterprise security resources, not only because it had not been seen in the wild before this attack, but also because an environment would have to be compromised already in order for this attack technique to be deployed. It's also worth noting that SAML is 14-year-old technology. We have counseled our customers to utilize more modern authentication techniques, or to move authentication (and other workloads) to the cloud to maximize security. But many customers use applications in their on-premises environments that rely upon SAML for authentication, so they continue to need us to support SAML in our on-premises authentication products.

The important lesson of Solorigate is that enterprises and government agencies need to improve basic cybersecurity hygiene measures that protect against the most common forms of attack. Those who want the best security should move to the cloud, where advanced security analytics can be provided at scale. The technology industry must work to provide better tools and systems to identity attacks, and Microsoft is working on new ideas to help customers detect sophisticated attacks like this one. The intelligence community will also have a role in helping to identify and defend against the most likely and impactful nation-state attacks against U.S. government infrastructures. And we need to establish internationally enforceable rules of nation state conduct in cyberspace to prevent the most disruptive and heinous forms of attack, including attacks on the software update processes, the primary means by which the industry keeps its technology secure and which customers must be able to trust.

6. **Has any Microsoft penetration testing exercise in the past five years revealed a weakness in Microsoft identity products which could have permitted the creation of new identities, changed user permissions, or led to similar behavior as that observed of the threat actor responsible for the Solorigate incident? If so, what was done to remediate the identified weakness?**

Penetration testing conducted on Microsoft's identity product code over the last five years has not revealed any security vulnerability that could directly lead to identity manipulation. A penetration test that relies on implementation issues specific to an operating environment, including implementation issues revealed by Solorigate, to escalate or modify account privileges or create a new account would not necessarily reveal a risk that can be mitigated by Microsoft identity products. The attacker in the Solorigate incident escalated privileges in customer environments and in a minority of cases managed to compromise the customer's cryptographic secrets used to digitally sign proofs of identity. It is possible for customers to prevent removal of these master secrets from their environments with the use of a Hardware Security Module, as recommended by Microsoft guidance as well as National Security Agency guidance, or through administrative hardening.

According to research published by CrowdStrike, the adversary inserted the first stage of its malware into SolarWinds' software by compromising the company's build server. This server turns the human-readable code written by programmers into code that can be installed on computers. One promising cybersecurity defensive technology to protect against this method of compromise is known as reproducible builds. In short, this technology guarantees that the same code, no matter on which server it is built, will produce the same output. Thus, by comparing the output from several independent build servers, a backdoor inserted by one compromised build server would be easily detected, as the output would differ from the other build servers.

7. **Please summarize Microsoft's efforts to date to support reproducible builds.**

Microsoft has been pursuing and continues to pursue efforts to support reproducible builds and to address challenges that impact our and others' ability to do so. Generating reproducible builds requires starting with the same source code and other inputs along with consistency across many aspects of the software configuration of the build environment. All of the actions that occur during a build process need to be deterministic and independent of variables that modify the output. Key activities include reading the exact source files used, performing preprocessing with tools, and using a compiler to transform and translate the preprocessed files into a form that can be installed on a computer.

Many conditions, actions, or aspects of input can disrupt the reproducibility of the processes that occur on a build server. A simple example is when a tool inserts the current time into data consumed in an intermediate build step. The consequence will be that the final output then contains a time-based variable. This variable causes the build process to not be reproducible on the same build server as well as on other build servers.

Microsoft recognizes the importance of reproducible builds for software assurance and has publicly acknowledged working towards generating product builds in a reproducible way. In January 2018, we explained that date and time stamps appear nonsensical in some Windows 10 components because setting the timestamp to be a hash in the resulting binary preserves reproducibility.[15]

Language and compilers can also hamper reproducibility. A compiler is a key tool used on build servers to transform human readable code into code that can be installed on computers. If a compiler behaves in a predictable and deterministic way, then it will generate the same output given the same inputs. A deterministic compiler is necessary to support reproducible builds.

Microsoft compilers for C#,[16] Visual Basic,[17] and F#[18] support a compiler option to generate deterministic assemblies. The footnoted feature descriptions identify a list of variables that can

[15] Why are the module timestamps in Windows 10 so nonsensical? | The Old New Thing (microsoft.com)
[16] C# Compiler Options - code generation options | Microsoft Docs
[17] Compiler Options Listed Alphabetically - Visual Basic | Microsoft Docs
[18] Compiler Options - F# | Microsoft Docs

impact reproducibility in addition to the source code itself. For example, different compiler versions can introduce changes in the build output. The deterministic compiler feature for C# and Visual Basic has been available for almost five years and for F# for almost four years. Deterministic build options are not available for other Microsoft compilers (e.g., for the programming language C, assembly, etc.).

Challenges with reproducible builds also exist when software includes external dependencies, which can change as their project's community adds new features and patches vulnerabilities.

To assist developers in locating the exact source files used in a build process, a project called Source Link[19] was transitioned to the .NET Foundation[20] in November 2017. Source Link is supported by Microsoft, and it allows metadata about source control management systems (whether for open-source or propriety software) to be inserted into the output of the build process, helping developers and verifiers trace back the exact source code that was used to generate a build. This is beneficial for general debugging and for making builds reproducible. Microsoft source control products, including GitHub Enterprise, Azure Repos and Azure DevOps Server, all work with Source Link. In February 2021, Microsoft posted guidance on how to generate reproducible builds using Source Link.[21]

8. **Does Microsoft recommend that the U.S. government support efforts to ensure that widely used open source software can be built in a reproducible manner? Does Microsoft have any other recommendations for what the U.S. government should do to increase the security of widely used open source software?**

Yes. Microsoft recommends that the U.S. government support efforts focused on enabling widely used open source software to be built in a reproducible manner, recognizing that realizing that outcome will take significant time – likely several years. Reproducible builds require the entire tool chain involved in the build process to support reproducibility. Example workstreams include keeping track of build inputs (especially the exact source files used in a build), making revisions to source code to remove characteristics that make it incompatible with reproducible builds, and improving available build tools and infrastructure (especially compilers for popular programming languages). Open source software developers use a wide array of languages and tools, and ensuring the entire tool chain supports the efforts of diverse, often community-based projects will be foundational to enabling developers to more seamlessly make changes to support reproducible builds. An informative example of a multi-year effort to enable a large open source project to be built in a reproducible way is here: ReproducibleBuilds - Debian Wiki.[22]

---

[19] GitHub - dotnet/sourcelink: Source Link enables a great source debugging experience for your users, by adding source control metadata to your built assets

[20] https://dotnetfoundation.org/

[21] GitHub - clairernovotny/DeterministicBuilds: Shows how to do deterministic builds with .NET

[22] ReproducibleBuilds - Debian Wiki

Microsoft also encourages the U.S. government to support open source security and broader software security foundations and projects, including the Open Source Security Foundation (OpenSSF),[23] which is carrying forward the work[24] of the Core Infrastructure Initiative and[25] the Open Source Security Coalition,[26] as well as SAFECode[27] and the OWASP Foundation.[28] OpenSSF is particularly focused on improving the security of widely used open source software. For example, this OpenSSF publication, "Threats, Risks and Mitigations in the Open Source Ecosystem,"[29] includes recommended mitigations for improving the security of the open source ecosystem. To further increase the security of widely used open source software, also consider the following guidance: practices for reducing risk when using open source: Microsoft Open Source Software Security;[30] and GitHub code security guidance: Code security - GitHub Docs.[31]

9. **Does Microsoft believe it would be worthwhile for the U.S. government to require that all new software created by or for the U.S. government be capable of being built in a reproducible manner?**

Requiring some software that is created by or for the U.S. government to be built in a reproducible manner would be worthwhile, especially if pursued in a manner reflective of operational challenges and limitations. Factors to be considered include the timing for such a requirement, the availability of support where implementation might be especially challenging (e.g., some open source projects, programming languages without deterministic compilers, etc.), and the ability to appropriately prioritize against other software assurance efforts.

Reproducible builds do increase assurance for a specific step in the software development process, but they do not by themselves guarantee secure software. For example, if the source code contains a vulnerability, then performing a reproducible build and verifying it by repeating the same process in another environment would not identify the vulnerability. Microsoft encourages the U.S. government to use various tools, including its procurement power as well as its research and development, standardization, and center of excellence capacities, to help strengthen software security. Reproducible builds should be considered within a risk-based context as one of numerous technologies that have the potential to increase software assurance and reduce cybersecurity risk. The process and timeline for when such a reproducible build requirement would be feasible, and for what software, merit study, the results of which could

[23] Home - Open Source Security Foundation (openssf.org)
[24] Technology and Enterprise Leaders Combine Efforts to Improve Open Source Security - Open Source Security Foundation (openssf.org)
[25] Home - Core Infrastructure Initiative
[26] GitHub - Open-Source-Security-Coalition/Open-Source-Security-Coalition
[27] Home - SAFECode
[28] OWASP Foundation | Open Source Foundation for Application Security
[29] wg-identifying-security-threats/Threats, Risks, and Mitigations in the Open Source Ecosystem - v1.1.pdf at main · ossf/wg-identifying-security-threats · GitHub
[30] Microsoft Open Source Software Security
[31] Code security - GitHub Docs

lead to worthwhile and deliberative action. The study could take into account possible benefits as well as unintended consequences of such a requirement, such as potentially limiting the diversity of platforms, open source libraries, languages, and toolsets that information and communications technology vendors would be able to use as a result. It could also explore how the U.S. government can help foster ecosystem readiness for such a requirement or advance other software assurance practices that reduce risk.