Cyber Intel Brief

Log4Shell: What We Know Now

See Threats. Stop Breaches. Together

*Prepared by **deepwatch Threat Intel Team***

Table of Contents

Log4Shell

Appendices & Feedback

Executive Summary

On December 9, security experts from Alibaba's Cloud Security Team discovered a zero-day vulnerability, designated CVE-2021-4428, in Apache's Log4j Java library, a logging application that is part of the Apache Logging Services project. The vulnerability has been dubbed "Log4Shell" due to the remote code execution capabilities. The vulnerability was assigned a severity score of 10 and affected all versions from 2.0-beta9 to 2.14.1.

Before the official CVE designation, proof-of-concept code (PoC) was publicly released. Shortly after, the first signs of scanning activity were observed to identify vulnerable systems. Since the initial PoC was disclosed, security vendor Check Point detected over 60 variations of the exploit.

After the initial release of our "*Customer Advisory: Significant Cyber Event*" on December 10, the deepwatch Threat Intel Team learned that multiple sources were reporting the active exploitation of Log4Shell. In response to these new developments, deepwatch released and enabled a new alert to detect the log4j vulnerability for customers. In addition, deepwatch Squads and Threat Operations teams worked with customers to mitigate the security risks regarding this vulnerability.

In the days following initial reporting of the vulnerability, numerous security vendors began publishing their observations of active exploitation, some were observed as early as December 1. Initially, most of the activity revolved around coinminer operations that quickly ramped up to include botnets and a previously undisclosed ransomware group known as Khonsari. Microsoft also observed the first signs of Nation-state activity exploiting Log4Shell, ranging from development testing to active exploitation.

On December 14, the original fix for CVE-2021-44228 was found to be insufficient by security researchers when they discovered that the fix allowed for denial of service attacks under certain circumstances and tracked it as CVE-2021-4506. This new update introduced another denial of service vulnerability, tracked as CVE-2021-45105.

The near-ubiquitous presence of Log4j across enterprises and the difficulty in identifying and patching systems that use it give threat actors a prime initial access vector to conduct follow on operations. It is expected that threat actors will continue to exploit Log4Shell to gain access to targeted organizations to drop ransomware, Cobalt Strike, various malware, botnets, and cryptomining payloads.

Log4Shell

Log4Shell: What We Know Now

KEY POINTS:

- Apache Log4j is a Java-based logging application and is part of the Apache Software Foundation's Apache Logging Services project.
- Alibaba Cloud Security Team revealed a zero-day vulnerability, tracked as CVE-2021-4428, and dubbed "Log4Shell," involving arbitrary code execution in Log4j 2.
- The deepwatch Threat Intel Team assesses with high confidence that threat actors will likely continue exploiting Log4Shell. Therefore, all organizations are highly encouraged to update Log4j to 2.17.0 as soon as possible.

What are Log4j and Log4Shell?

Apache Log4j is a Java-based logging application that is part of the Apache Software Foundation's Apache Logging Services project.

On December 9, security researchers with Alibaba's Cloud Security Team discovered a zero-day vulnerability, tracked as CVE-2021-4428. This vulnerability could allow threat actors the ability to conduct arbitrary code execution in Log4j 2, with the description "Log4Shell."

What Software Products are Vulnerable to Log4Shell?

CISA has published a GitHub repository, "log4j-affected-db," which lists official CISA guidance and resources, current activity alerts, mitigation guidance, and a software list. In addition, CISA has a disclaimer that states, "*The information in this repository is provided "as is" for informational purposes only and is being assembled and updated by CISA through collaboration with the broader cybersecurity community.*"

First Patch Released and Follow-on Updates

The Apache Software Foundation issued an emergency security update on December 10 regarding the Java library Log4j after releasing the PoC and reports of active scanning for vulnerable servers. This vulnerability affects all versions from 2.0-beta9 to 2.14.1 with a severity score of 10.0 on the CVSSv3 severity scale.

On December 14, Apache released a security update stating that in some non-default configurations, the fix for CVE-2021-44228 in Apache Log4j 2.15.0 was incomplete and posed a new vulnerability, tracked as CVE-2021-45046 with a severity score of 3.7 on the CVSSv3 severity scale. Days later, on December 17, the CVE severity score was raised from 3.7 to 9.0 after security researchers discovered that in certain situations, threat actors could have *"remote code execution in some environments, and local code execution in all environments."* Of note is that *"remote code execution has been demonstrated on macOS but no other tested environments."*

On December 18, Apache released another security update stating that in some non-default configurations, the fix for CVE-2021-45046 in Apache Log4j 2.16.0 was incomplete and posed a new vulnerability, tracked as CVE-2021-45105 with a severity score of 7.5 on the CVSSv3 severity scale. Additionally, in Apache's advisory, they state that *"Apache Log4j2 versions 2.0-alpha1 through 2.16.0 did not protect from uncontrolled recursion from self-referential lookups. When the logging configuration uses a non-default Pattern Layout with a Context Lookup. Attackers with control over Thread Context Map (MDC) input data can craft malicious input data that contains a recursive lookup, resulting in a StackOverflowError that will terminate the process (Denial-of-Service)."*

Log4Shell

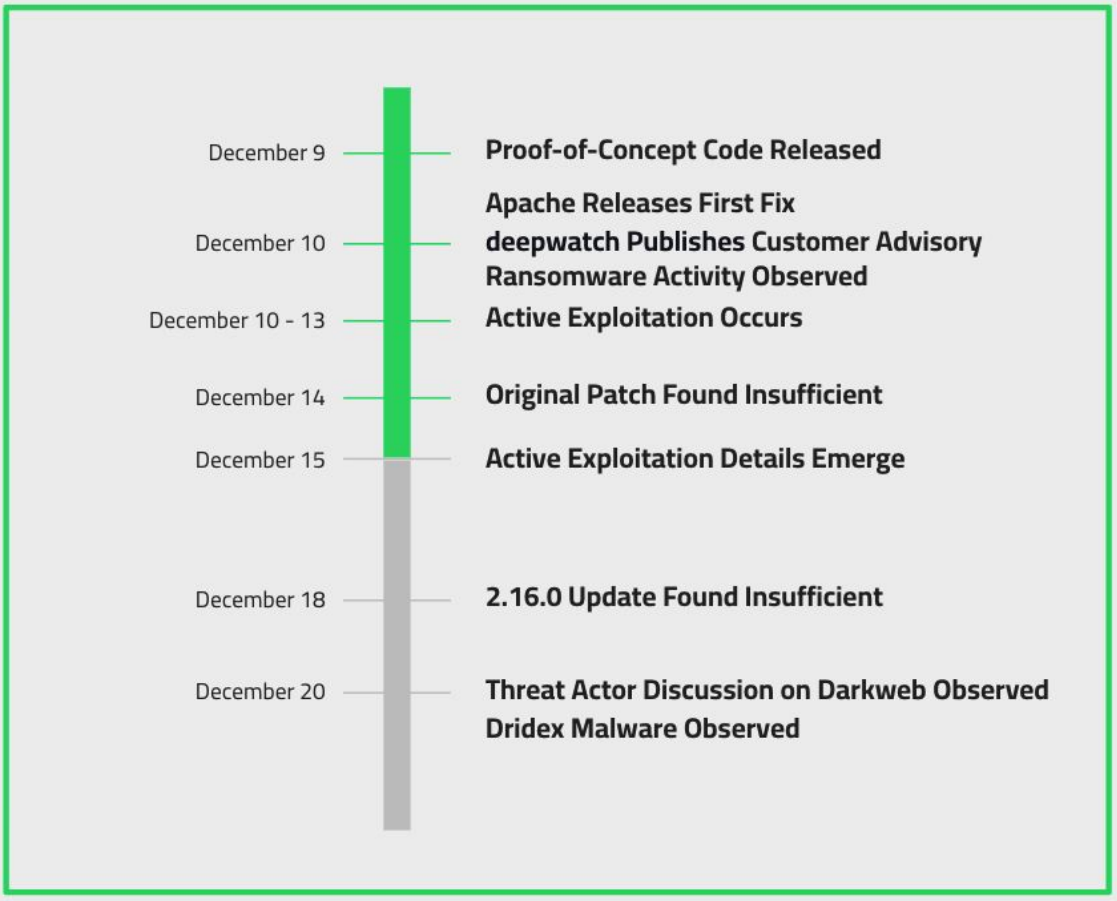
Log4Shell: What We Know Now

What Mitigation Measures Should I Take?

As of December 21, Apache's official mitigation guidance states that organizations running Log4j 1.x are not impacted by CVE-2021-45105 or CVE-2021-45046. Apache recommends organizations running Log4j 2.x should Implement one of the following mitigation techniques:

- Java 7 users should upgrade to release 2.12.2.
- Java 8 (or later) users should upgrade to release 2.17.0.
- Alternatively, this can be mitigated in the configuration file:
    - In PatternLayout in the logging configuration, replace Context Lookups like ${ctx:loginId} or $${ctx:loginId} with Thread Context Map patterns (%X, %mdc, or %MDC).
    - Otherwise, in the configuration, remove references to Context Lookups like ${ctx:loginId} or $${ctx:loginId} where they originate from sources external to the application such as HTTP headers or user input.
        - Note that only the log4j-core JAR file is impacted by this vulnerability. Applications using only the log4j-api JAR file without the log4j-core JAR file are not impacted by this vulnerability.
- Also, note that Apache Log4j is the only Logging Services subproject affected by this vulnerability. Other projects like Log4net and Log4cxx are not impacted by this.

Timeline of Events



Log4Shell

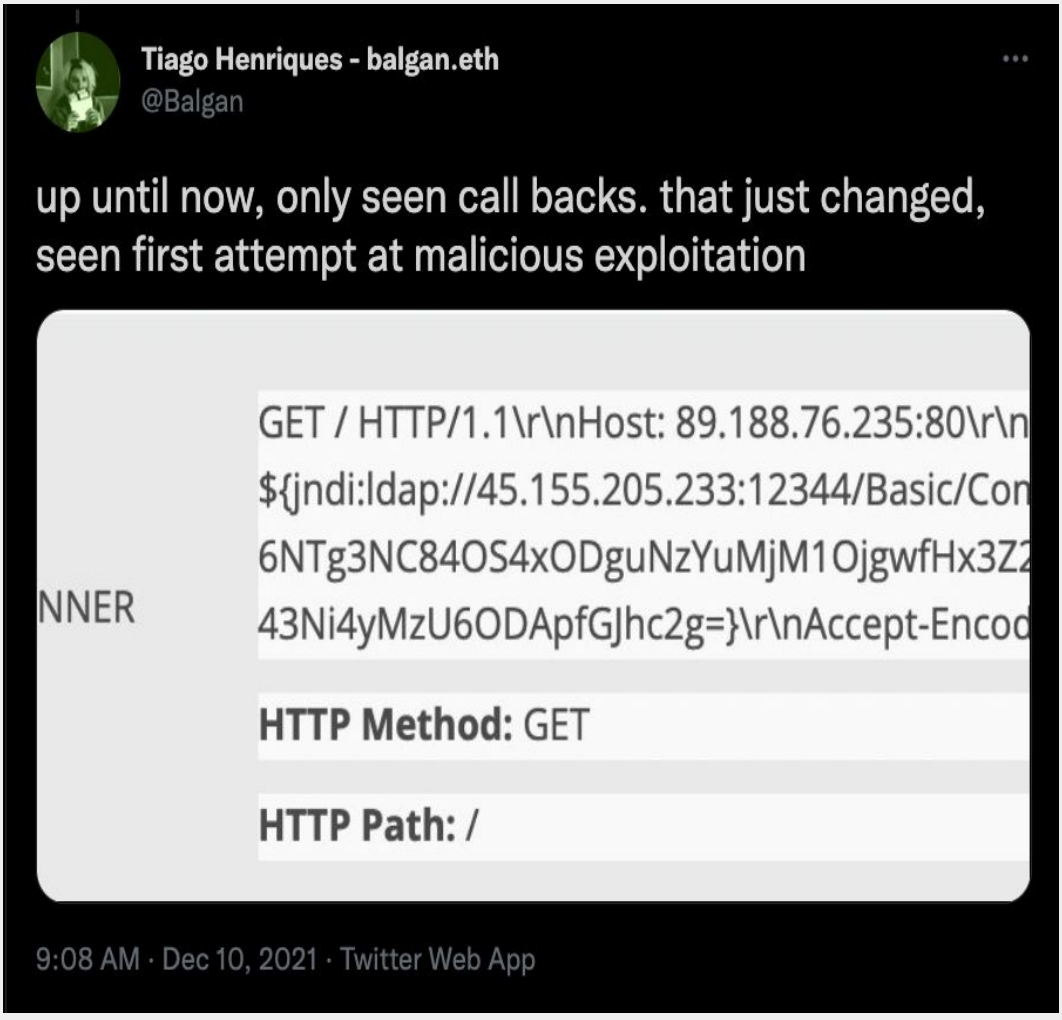
Log4Shell: What We Know Now

Proof-of-Concept, Darkweb, and Active Exploitation

Before the CVE identification was given, the initial proof-of-concept (PoC) for CVE-2021-44228 was disclosed on December 9 on GitHub. Since then, additional PoCs have been released on GitHub. On December 12, Check Point detected over 60 different exploits of the original variation.

Because CVE-2021-4428 is a critical severity and publicly available PoC deepwatch issued a Customer Advisory: Significant Cyber Event (CA: SCE).

Shortly after the initial release of our CA: SCE, the deepwatch Threat Intel Team learned that multiple sources were reporting the active exploitation of CVE-2021-44228. In response to these new developments, deepwatch released and enabled a new alert to detect the log4j vulnerability for customers. deepwatch Squads and Threat Operations teams worked with customers to mitigate the security risks regarding this vulnerability.



Source: Twitter

Log4Shell

Log4Shell: What We Know Now

Numerous security vendors began publishing their observations regarding threat actor activity within days of deepwatch's CA: SCE release.

Bitdefender, on December 10, observed various attacks on their honeypots, as well as real-world attacks on machines running Bitdefender's endpoint protection agent. Additionally, Bitdefender observed the Muhstik botnet, XMRIG miner, Khonsari, a new ransomware family, Orcus RAT, and reverse bash shell activity as follow-on activity.

Microsoft Threat Intelligence Center (MSTIC) published on December 11 their observations on Log4Shell, stating, "*vulnerability being used by multiple tracked nation-state activity groups originating from China, Iran, North Korea, and Turkey. This activity ranges from experimentation during development, integration of the vulnerability to in-the-wild payload deployment, and exploitation against targets to achieve the actor's objectives.*"

Cloudflare and Cisco Talos stated that they had observed exploitation attempts before the official public notification of CVE-2021-4428, and any publicly available PoC was available. Mathew Prince, CEO of Cloudflare, posted the following tweet:

Source: Twitter

In an update to their blog post published on December 10, Cisco Talos stated that the earliest observed threat actor activity exploiting CVE-2021-44228 started on December 2.

Check Point stated in a blog post published December 12, that, "*Since we started to implement our protection we prevented over 3,700,000 attempts to allocate the vulnerability, over 46% of those attempts were made by known malicious groups. We have so far seen an attempted exploit of 48% of corporate networks globally.*"

Sophos disclosed their observations on December 12, stating, "*Sophos is already detecting malicious cryptominer operations attempting to leverage the vulnerability, and there are credible reports from other sources that several automated botnets (such as Mirai, Tsunami, and Kinsing) have begun to exploit it as well.*"

The Swiss CERT published an advisory stating that *"The exploitation attempts observed by us so far were used to deploy mass-malware like Mirai, Kinsing and Tsunami (aka Muhstik). The primary use of these botnets is to launch DDoS attacks (Mirai, Tsunami) or to mine cryptocurrencies (Kinsing)."*

Log4Shell

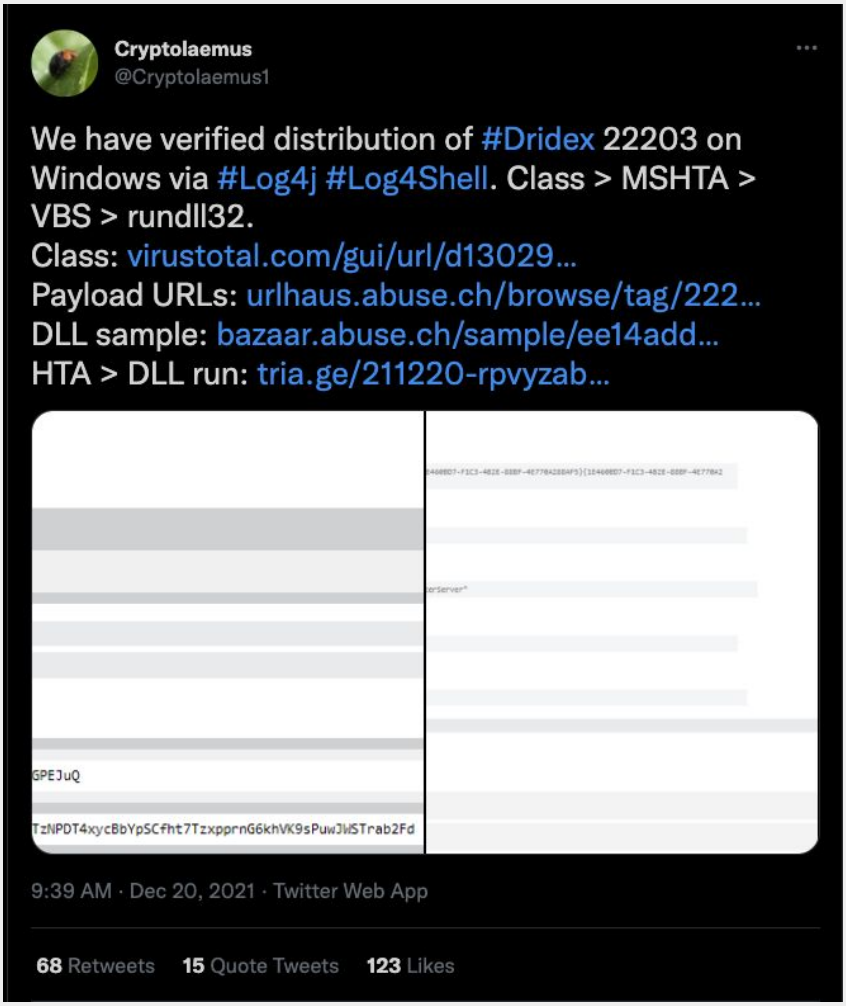
Log4Shell: What We Know Now

In a blog post published on December 13, Trend Micro observed post-exploitation payloads of the Mirai botnet and Kinsing coinminer.

In an update on December 15 to Check Point's blog post that they *"can report that a known Iranian hacking group (commonly associated with the local regime), named "Charming Kitten" or APT 35, is also behind an attempt to exploit the Log4j vulnerability against 7 targets in Israel (from the government and business sector) in the last 24 hours."* Further confirming Microsoft's observations published on December 12.

The intelligence firm Advanced Intel (AdvIntel), on December 17, released a blog post that the ransomware group Conti was using Log4Shell as an initial access vector and lateral movement targeting VMWare vCenter. Additionally, AdvIntel's intelligence collection operations observed on December 12 that some members of the Conti group were interested in exploiting the vulnerability for the initial attack vector, culminating in scanning using the publicly accessible Log4J2 exploit.

On December 20, the research group Cryptolaemus posted a tweet that confirmed that threat actors were using Log4Shell to drop the banking trojan Dridex. Once a malware that stole banking credentials, the trojan has since evolved to be a loader that downloads various modules that can be used to install additional payloads, infect other machines, take screenshots, as well as other malicious behaviors.



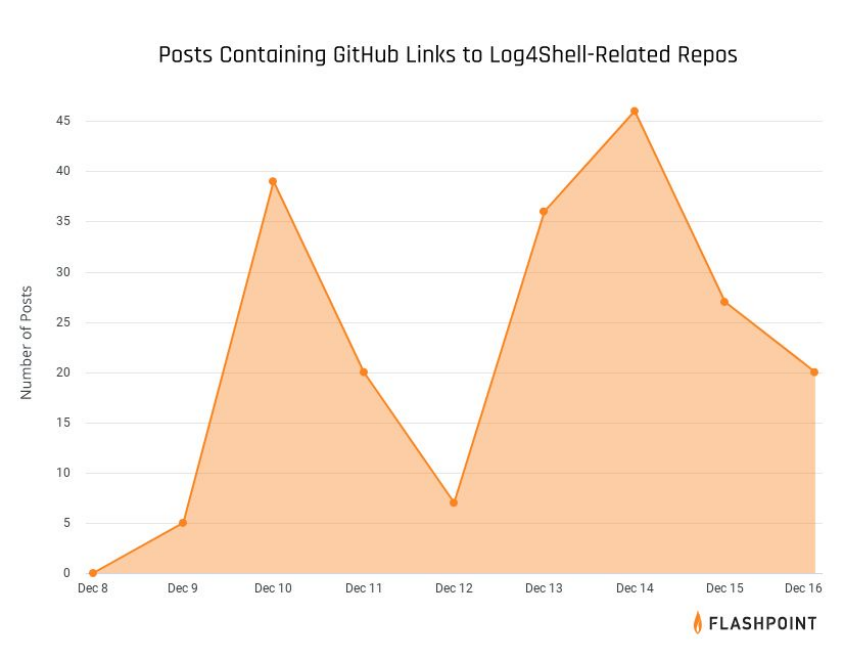
Source: Twitter

Log4Shell

Log4Shell: What We Know Now

The intelligence firm Flashpoint has monitored threat actor discussions on criminal dark web forums XSS, Raid, and RAMP and published their findings on December 20. For example, on the top-tier Russian-language hacking forum XSS, Flashpoint observed threat actors sharing PoCs, payloads to evade web application firewalls, contributions to mapping out the Log4Shell attack surface, how-tos, and patch updates from Apache. Also, on XSS, Flashpoint has observed threat actors sharing cloned Log4Shell PoC GitHub repositories. These cloned repositories are archived and remain available even if GitHub removes the cloned repositories and act as a knowledge base for threat actors.

To avoid law enforcement actions, the admins of the forum RAID have been removing posts related to Log4Shell. But threat actors are downloading PoCs exploits and custom Log4Shell scanning tools and plugins for well-known vulnerability scanners specifically designed to identify systems vulnerable to Log4Shell before the admins can delete the posts.



Data derived from Flashpoint's collections, including chatter on various illicit communities. Source Flashpoint.

deepwatch Threat Intel Outlook:

The Threat Intel Team assesses with high confidence that multiple Threat groups will employ exploitation attempts across all Internet-facing systems. Many Threat groups will utilize this critical and widespread vulnerability to install malicious payloads to gain initial access to systems and conduct lateral movement. These payloads could be cryptomining software, backdoors like WebShells, and Post-Exploitation Command and Control (C2) payloads like Cobalt Strike ultimately leading to sensitive data exfiltration or ransomware attacks. The Threat Intel Team strongly urges all organizations to review their Internet Facing systems to identify vulnerable systems to "Log4j" CVE-2021-44228 & CVE-2021-45046 and take immediate risk reduction efforts by updating to 2.16.0 as soon as possible. If you cannot update at this time, it is recommended that organizations follow the mitigation steps in Apache's security update advisory.

Appendices

Appendix A

Purpose

This report is provided to you to improve your situational awareness and educate recipients of cyber events to aid in protecting organizations' networks, proprietary and personally identifiable information from unauthorized access, theft, or espionage. In addition, deepwatch includes additional insights and recommendations and any actions we may have taken if applicable.

Sources of Information

This publication incorporates open-source news articles to educate readers on cybersecurity matters IAW USC Title 17, section 107, Para a. All articles are truncated to avoid the appearance of copyright infringement.

Use and Definitions

To help you use this document to its full potential, a few items may be helpful to know:

- You can click on any item in the table of contents to take you to that portion of the report.
- Links throughout this document are identified by the font color of "deepwatch" Green.
- Each headline will be categorized; these categories quickly let you know what the main topic of the headline is.
- **Each headline this report covers includes the following information:**
    - A headline, publication date, and a link to the source material.
    - Key Points section to get the most important information first.
    - Summary - This is a brief synopsis of the reporting to bring you only the most relevant information. If applicable, deepwatch will link items of interest for further context; these will be in "deepwatch" Green.
    - deepwatch Threat Intelligence Outlook: - This section may include additional analysis and reporting on the activity if applicable, any recommendations, and any actions deepwatch may have taken with the available information.

Appendix B

What We Mean When We Say: An Explanation of Estimative Language

To convey analytical assessments and judgments, the Threat Intel Team uses phrases like judge, assess, and estimate, as well as probabilistic terms like probably and likely. Such claims are not based on facts, proof, or knowledge. These evaluations and judgments are frequently based on gathered data that is often incomplete or fragmentary. Some evaluations are based on prior judgments. In all cases, assessments and judgments are not meant to imply that we have "proof" that something is a fact or that two items or issues are inextricably linked.

in addition to conveying judgments rather than certainty, our estimative language frequently conveys 1) our assessed likelihood or probability of an event; and 2) the level of confidence we attribute to the judgment.

Estimates of Likelihood. We use probabilistic language to reflect the Intel Team's estimates of the likelihood of developments or events because analytical judgments are not certain. Terms like "probably," "likely," "very likely," and "almost certainly" denote a higher than even chance. The terms unlikely and remote imply that an event has a lower than even chance of occurring; they do not imply that it will not. Terms like might and might reflect situations where we are unable to assess the likelihood, usually due to a lack of relevant information, which is sketchy or fragmented. Terms like "we can't dismiss," "we can't rule out," and "we can't discount" refer to an unlikely, improbable, or distant event with significant consequences.

Confidence in Assessments. Our assessments and projections are based on data that varies in scope, quality, and source. As a result, we assign our assessments high, moderate, or low levels of confidence, as follows:

- High confidence indicates that our decisions are based on reliable information and/or that the nature of the problem allows us to make a sound decision. However, a "high confidence" judgment is not a fact or a guarantee, and it still carries the risk of being incorrect.
- Moderate confidence denotes that the information is credible and plausible, but not of high enough quality or sufficiently corroborated to warrant a higher level of assurance.
- Low confidence indicates that the information's credibility and/or plausibility are in doubt, that the information is too fragmented or poorly corroborated to make solid analytic inferences, or that we have serious concerns or problems with the sources.

Feedback

Please take a few minutes to send us your feedback here. Your feedback submission may be anonymous. We read each submission carefully, and your feedback will be extremely valuable to the deepwatch Threat Intel Team. Feedback should be specific to your experience with this written product to enable deepwatch to make quick and continuous improvements to these products.