WHITE PAPER

BROADCOM

BLAST-RADIUS Patching for VIP Enterprise Gateway

Handling a Protocol Upgrade

TABLE OF CONTENTS

Identify the Problem

Due to a recently discovered vulnerability, vendors have initiated a necessary patching cycle. While patching to address vulnerabilities is not a new occurrence, this particular patch also forces a change to the RADIUS communications protocol. The RADIUS upgrade offers a more secure set of transactions.

Upgrading RADIUS might not be so straightforward, as two or more sets of software might need to be upgraded at the same time: one set of software that acts as the RADIUS client and one set that acts as the RADIUS server—*both* need to be upgraded to communicate using the more secure RADIUS. This white paper examines how to address these upgrades in a production environment, without incurring a significant outage.

The Root of the Problem

RADIUS uses UDP, a connectionless networking protocol that essentially means a server waits for incoming packets. When a client occasionally sends a packet, the server needs to react quickly, process that packet, and reply back to the client. For many RADIUS clients this process completes within 30 to 60 seconds, but that timespan can be longer.

At authentication time, the RADIUS client sends an Access-Request. If the server has all it needs, it sends back the appropriate Access-Accept or Access-Reject. If this is a longer transaction that both the client and the server support, there will be an additional leg to the transaction through the Access-Challenge, and then the client sends a new Access-Request and the server delivers a final disposition, either Access-Accept or Access-Reject.

The main takeaway here is that the RADIUS client and server are both needed during the communication process and there is no “What language do you speak?” moment—they either understand each other or they don’t. Incorrect communication is silently discarded, failing authentication. Refer to the RADIUS RFC for more information.

Investigate the Problem

WE NEED SOMETHING MORE THAN PROTOCOL SUPPORT OR NON-SUPPORT — WE NEED A SOLID MIGRATION STRATEGY

If we upgrade only the RADIUS servers to communicate differently, the RADIUS clients might fail. If we upgrade only the RADIUS clients to communicate differently, the RADIUS servers might fail. If all developers followed Postel's Law (the robustness principle), we wouldn't need to worry about this, as unknown parameters would simply be ignored. But this isn't happening in real life. Some implementations that receive the new attribute are failing. Some implementations that don't receive the new attribute might fail. With so many RADIUS client and server implementations, it's realistic to expect this variety of outcomes. Aside from having further justification for a test environment, we need something more than protocol support or non-support—we need a solid migration strategy.

Robustness = Computer Hope

VIP Enterprise Gateway implemented two new modes:

- Compliant – The desired future state
- Compatible – A less strict intermediate state

As articulated in a detailed KB article, the patched VIP Enterprise Gateway will always use the `Message-Authenticator` attribute. It just differs over whether it expects the RADIUS client message to do so.

This is good, and may even be enough, but can it be better? The pre-patched VIP Enterprise Gateway would never send a `Message-Authenticator` attribute, but you could send it packets with and packets without `Message-Authenticator`, and VIP Enterprise Gateway would still handle them. Not only is this an example of the robustness principle, but this should also be enough for a plan: one side of the communication provides enough flexibility to handle all the scenarios that we need.

The Test

BEFORE WE CAN FORM A MIGRATION PLAN, WE NEED INSIGHT INTO WHAT KIND OF FAILURES WE HAVE ON OUR HANDS

We can't assume that all participants will communicate properly. Some may fail upon receiving the `Message-Authenticator` attribute. Some will fail without it. Worse, after old applications are upgraded, the patches in those systems may introduce failures. Before we can form a migration plan, we need insight into what kind of failures we have on our hands.

Test for Failures

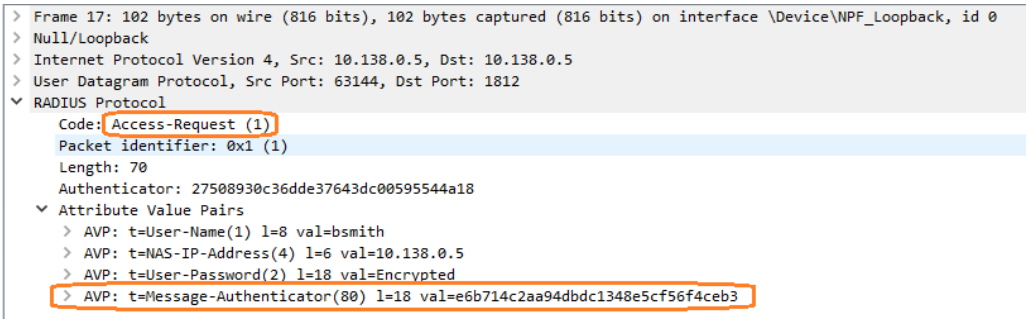
To identify failures, complete the following steps in a test environment:

1. Replicate production and observe which RADIUS clients are already using the `Message-Authenticator` attribute. These can be excluded from further consideration.

   NOTE: This analysis only looks at the presence of the `Message-Authenticator` attribute and the issues that it causes. It does not examine the proper processing of the `Message-Authenticator` attribute. This should also be examined and is outside the scope of this discussion.

2. Upgrade VIP Enterprise Gateway. Now all RADIUS clients will receive RADIUS messages with the `Message-Authenticator` attribute. Do they handle authentication correctly, meaning are they robustly ignoring an unknown attribute? If any fail, note these as fragile apps that prefer legacy handling (no `Message-Authenticator` attribute). Also keep note of the apps that passed this test.

3. Revert VIP Enterprise Gateway to its pre-patch status and then upgrade any RADIUS clients so that they are now using the `Message-Authenticator` attribute. Confirm this with a traffic capture:



NOTE: VIP Enterprise Gateway will not be sending the `Message-Authenticator` attribute. Upgraded RADIUS clients should be silently rejecting these. They might have a compatibility mode that can be engaged (and tested in this case). Note which ones succeed and which ones fail. This can be a different list than before. Here we are looking for any fragile apps that prefer modern handling.

4. Upgrade VIP Enterprise Gateway and confirm that, with everything upgraded, communication works. VIP Enterprise Gateway should be in compatible mode up to this point. A completion of this test may engage the compliant mode to be sure VIP Enterprise Gateway handles communication correctly. Use an old testing utility to confirm that non-compliant traffic is both silently discarded by VIP Enterprise Gateway and does not produce faults in the running processes.

The Test Results

These tests help sort apps into four categories:

BEFORE WE CAN FORM A PLAN, WE NEED INSIGHT INTO FAILURES AND WHAT KIND OF FAILURES WE HAVE ON OUR HANDS

- **Robust:**
  - Pre-upgrade and post-upgrade, the app will correctly handle unexpected input.
- **Fragile Legacy:**
  - Pre-upgrade, the app will fail when it receives the `Message-Authenticator` attribute.
  - Post-upgrade the app will work correctly.
- **Fragile Modern:**
  - Pre-upgrade, the app will work correctly.
  - Post-upgrade, the app will fail to handle the absence of the `Message-Authenticator` attribute correctly.
- **Fragile Both:**
  - Pre-upgrade, the app will fail when it receives the `Message-Authenticator` attribute.
  - Post-upgrade the app will fail to handle the absence correctly.

At this point, we might have a list like the following:

- RobustApp1
- FragileLegacy2
- RobustApp3
- FragileModern4
- FragileLegacy5
- FragileBoth6
- RobustApp7

The Plan

In a worst-case situation, an upgrade strategy that seeks to minimize downtime would look something like this:

1. Confirm each app is configured to use two VIP Enterprise Gateways: EG1 and EG2.
2. Adjust the order that each app uses as follows:
   - Robust apps and Fragile Modern apps use EG2.
   - Fragile Legacy and Fragile Both apps use EG1 as their primary.
3. Upgrade EG2. Confirm that app behavior works as expected.
4. Upgrade any Fragile Both apps and change their configuration to use EG2 as primary before putting them back into service. Confirm these apps are working as expected.
5. Upgrade the Fragile Legacy apps. Confirm these are working as expected.
6. Upgrade EG1.
7. Optionally, engage compliant mode on both EGs.

Add additional VIP Enterprise Gateway servers as necessary for robustness during the upgrade. This ensures that applications are not sending traffic to a singular validation server, but that they have access to additional servers if issues arise or the load requires.

Future Thoughts

The industry's move to Zero Trust is reducing the need to have internal networks, which RADIUS assumes. MD5 protection on passwords for PAP is not good enough across the Internet, and neither is CHAP. Newer technologies, such as code flow in OIDC or SAML, function without the need for an internal network and are likely to replace RADIUS.

This evolution means that the need for a scramble due to low-level protocol issues is being replaced with compatibility and configuration issues at the low level, and protocol flexibility to handle changes such as passkey backups and the effect on relevant mandates such as PSD2 and EPCS, consent management, privacy management, and more.

VIP-BR-WP100 August 26, 2024