# CVE 2017-5689
# Manually Exploiting Intel AMT Vulnerability

Laxita Jain
*University of Delhi, Lucideus Technologies*
laxita.ac@gmail.com

---

***Abstract*: This document illustrates the manual exploitation of the vulnerability found in the Intel Active Management Technology in 2017 that stripped off the primary authentication mechanism in the Intel AMT web interface, conferring maximum i.e. admin privileges to an adversary from a remote location. A brief explanation of the web interface is given as well as the underlying hardware, following which, we explore the vulnerability and demonstrate how to test it.**

***Keywords***: *Intel AMT Vulnerability, Intel ME Vulnerability, Remote Hardware Access, Manual exploitation*

## I. Introduction

Intel AMT i.e. Intel Active Management Technology is a firmware that allows remote access to a computer's hardware. It is intended for system admins to remotely fix, manage, update certain business computers, desktops, servers, and even smart vending machines. This technology is largely used where either the employees are constantly on a move or the number of systems is huge; it eases the sysadmin's job to work out malfunctions in a computer, no matter where it is located, provided it has internet connectivity.

### A. Intel Management Engine

Select business computers with AMT have a tiny sub-computer installed, aside from the main microprocessor. It is called the Intel Management Engine. With full access to the computer's hardware, it can perform several tasks while the system is booting, running or even in sleep mode. It can help you remotely access the memory, keyboard, drivers, BIOS, and even the Network.

### B. Intel AMT Web Interface

The web interface can be accessed by port numbers:

16992: AMT Web Server with HTTP

16993: AMT Web Server with HTTPS

Logging onto the interface requires authentication. Here, the vulnerability bypasses this authentication.

## II. Understanding the Vulnerability

For authentication, the AMT Web Server uses Digest-Based Authentication mechanism where the passwords are converted to an encrypted string before being transmitted.

At the server, the user entered password is compared to the one that's already saved and stored. For this, a string comparison function is called which has the following usage:

strcmp(*user_given_string*, *stored_string, length_to_be compared*)

This function returns a +ve value or -ve value when the strings are different or a 0 when they're the equal/same. And hence, 0 implying that you're authenticated. The length is the number of characters to be compared, and here, it is the length of the user provided string.

But what happens when the length is zero? That implies that the number of characters to be compared is none. Is null equal to null? When there's nothing to compare, the function returns a 0. And that's the vulnerability.

If you use a proxy server (like Burpsuite) to intercept the request and alter it to send a null string instead of the password string, you are able to log into the interface as an authenticated user and have the entire control of the remote PC.

## III. Testing the Vulnerability

**Step 1**: Configure the proxy server to intercept requests. Open the Intel AMT interface using the IP address and port (can be found from shodan.io). An authentication dialogue box will pop up. Enter the User Name as "admin" and any random string as the password.

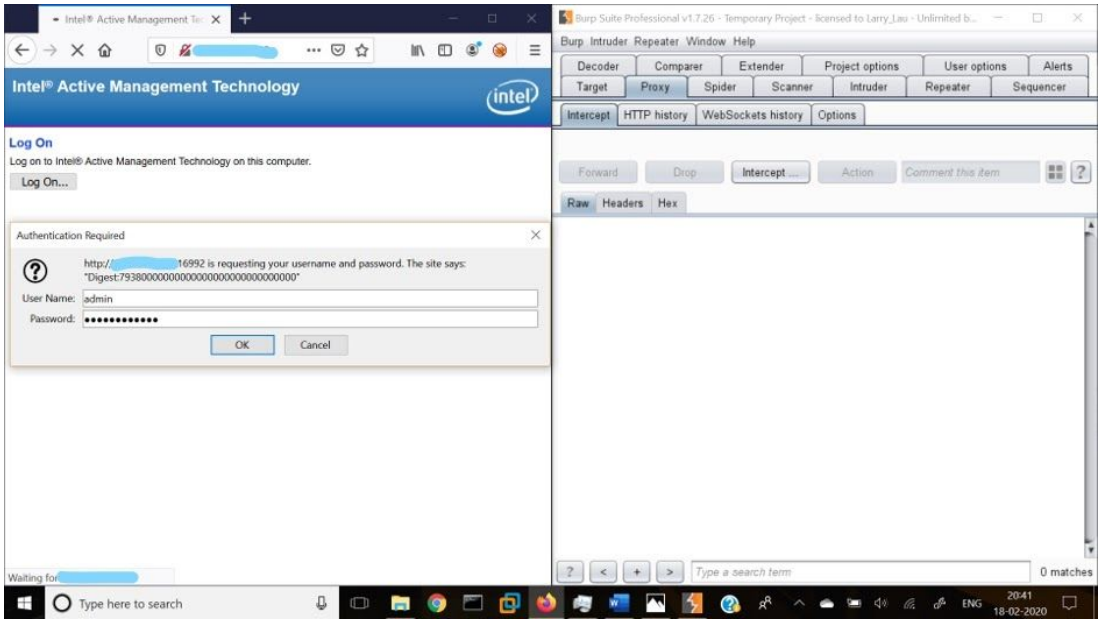


Fig. 1 Intel AMT web interface's authentication dialogue box

‘admin’ is the default administrator  username used in all the AMT interfaces; we need not even alter this in the request for authentication. However, entering a random string for password works because we edit this field in the intercepted request.

**Step 2:** Read the intercepted request and find the “response” parameter.

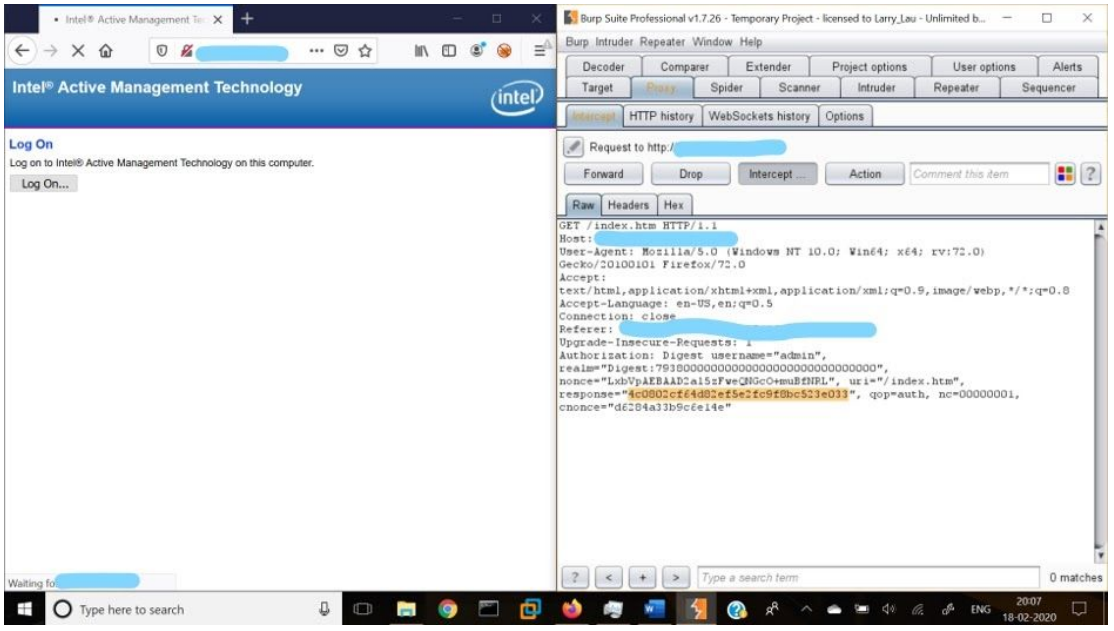


Fig. 2 Intercepted login request

The ASCII string we entered as the password gets converted into a hexadecimal string, observing the procedure of digest-based authentication. This string is sent in the “response” parameter.

**Step 3:** Clear the string value in the response parameter.

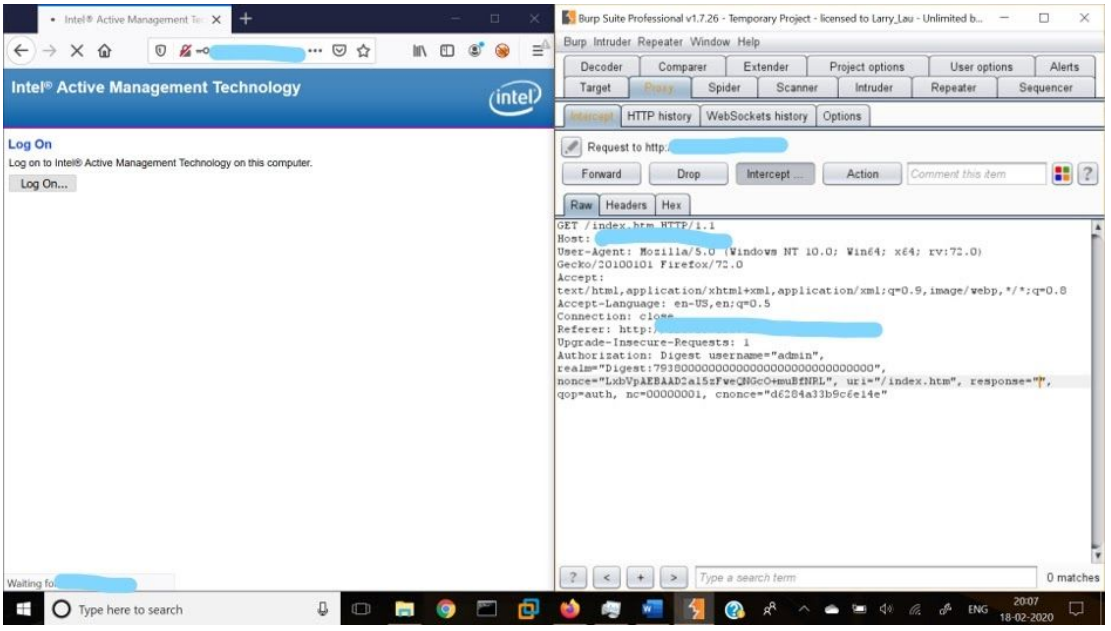


Fig. 3 Request with changed ‘response’ parameter value

Clearing the ‘response’ parameter sends to the server a ‘null’ string instead of the hexadecimal string value.

**Step 4:** Forward the request and you’re in!

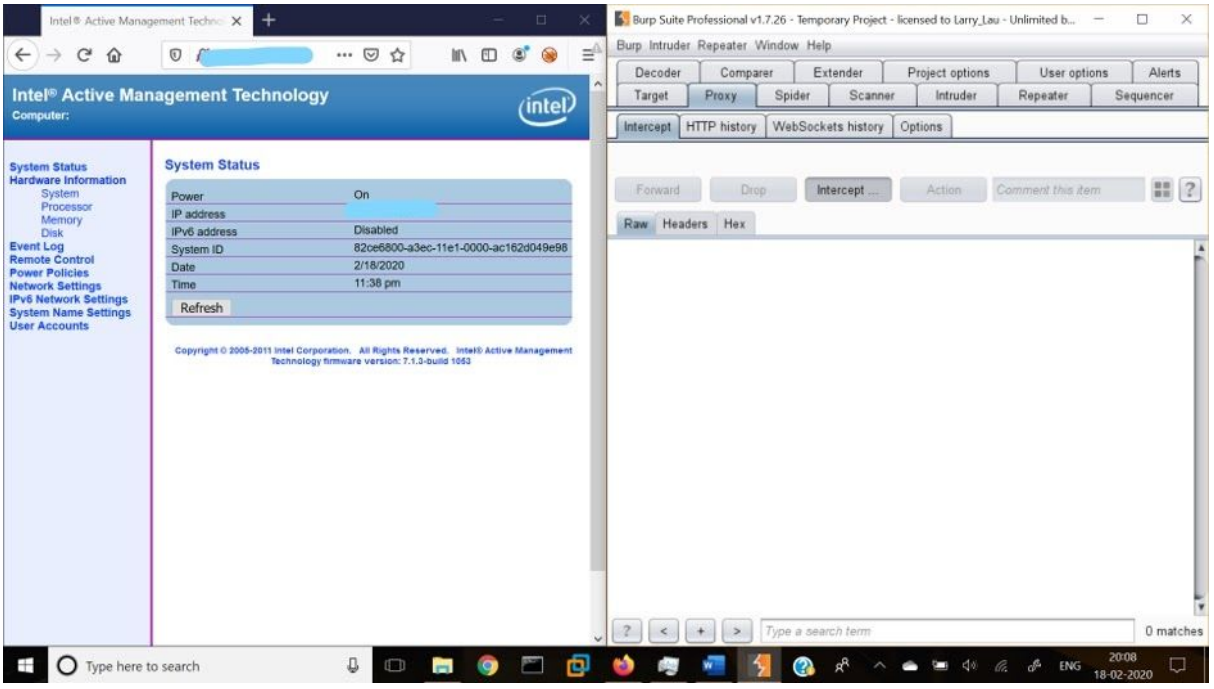


Fig. 4 Intel AMT web interface, logged in

The null or empty string sent to the server lets the comparison function take it as its input parameter and continue normally processing the defective request. It calculates the length as 0 (zero) and gives a green light of authenticity as the conditions are deceptively, but algorithmically satisfied.

## IV. Conclusion

CVE 2017-5689, Intel AMT vulnerability can be exploited in a very short span of time, and yet can be fatal for computer systems. Attackers can exploit this for doing extreme harm, physically damaging entire computer systems and networks. Since Intel ME still functions despite AMT being disabled, once in the network, an attacker can activate it in other systems and continue pivoting. The CVSS of this vulnerability is, undoubtedly, **9.8**.

## References:


- https://www.blackhat.com/docs/us-17/thursday/us-17-Evdokimov-Intel-AMT-Stealth-Breakthrough-wp.pdf
- https://nvd.nist.gov/vuln/detail/CVE-2017-5689