Introduction

This report is about the Memory Corruption vulnerability in Microsoft Internet Explorer 8,9,10 and 11 which allows remote attackers to execute arbitrary code or cause a denial of service via a crafted web site. This vulnerability is also known as "Internet Explorer Memory Corruption Vulnerability".

What is Memory Corruption?

In short it means that the software performs operations on a memory buffer but it reads from or writes to a memory location that is outside of the intended boundary of the buffer. Some languages allow direct addressing of memory locations and do not automatically ensure that these locations are valid for the memory buffer that is being referenced. This can cause read or write operations to be performed on memory locations that may be associated with other variables, data structures, or internal program data. As a result, an attacker may be able to execute arbitrary code, alter the intended control flow, read sensitive information, or cause the system to crash.

This example, written in C, takes an IP address from a user, verifies that it is well formed and then looks up the hostname and copies it into a buffer.

```c
void host_lookup(char *user_supplied_addr){
  struct hostent *hp;
  in_addr_t *addr;
  char hostname[64];
  in_addr_t inet_addr(const char *cp);

  /*routine that ensures user_supplied_addr is in the right format for conversion */
  validate_addr_form(user_supplied_addr);
  addr = inet_addr(user_supplied_addr);
  hp = gethostbyaddr( addr, sizeof(struct in_addr), AF_INET);
  strcpy(hostname, hp->h_name);
}
```
This function allocates a buffer of 64 bytes to store the hostname however there is no guarantee that the hostname won’t be larger than 64 bytes. If an attacker specifies an address which resolves to a very large hostname, then we may overwrite sensitive data or even abandon control flow to the attacker.

**Internet Explorer Memory Corruption Vulnerability**

It is a remote code execution vulnerability which means when Internet Explorer improperly accesses an object in memory, it could corrupt memory in such a way that an attacker could execute arbitrary code in the context of the current user. An attacker who successfully exploited these vulnerabilities could gain the same user rights as the current user. If the current user is logged as administrator, an attacker who successfully exploited these vulnerabilities could take complete control of an affected system. Attacker could then install programs; view, change, or delete data; or create new accounts with full user rights. Systems where Internet Explorer is used very often such as terminal servers or workstations are at the most risk from these vulnerabilities.

**Mitigating factors and workarounds**

In any case an attacker does not have the power to force users into visiting these specially crafted websites. Instead, an attacker would have to persuade users to visit the website usually by getting them to click on a link in an email message that takes users to the attacker's website. Usually all supported versions of Microsoft Outlook, Microsoft Outlook Express and Windows Mail open HTML email messages in the Restricted sites zone. The Restricted sites zone disables script and ActiveX controls. This helps to reduce the risk of an attacker being able to exploit this vulnerability.

There is another workaround that would help block known attack vectors. User should set Internet and Local internet security zone settings to "High" to block ActiveX Controls and Active Scripting in these zones, but by setting the browser’s security to high, user can also miss content made by safe websites to add functionality through ActiveX. There is also a middle option between allowing and disallowing all ActiveX. This means that when ActiveX wants to intervene then the user can decide for him- or herself whether she trusts the website or not.
Internet Explorer on Windows Server 2003, Windows Server 2008, and Windows Server 2008 R2 runs in a restricted mode that is also known as Enhanced Security Configuration. This mode mitigates this vulnerability.

This vulnerability was fixed in a security update 2925418 which changes the way Internet Explorer handles objects in memory.

References

1. [http://cwe.mitre.org/data/definitions/119.html](http://cwe.mitre.org/data/definitions/119.html)